USING MACHINE LEARNING AND MODEL INTERPRETATION AND VISUALIZATION TECHNIQUES TO GAIN PHYSICAL INSIGHTS IN ATMOSPHERIC SCIENCE

Amy McGovern
School of Computer Science
110 W Boyd St
Norman, OK 73019
amcgovern@ou.edu

Ryan A. Lagerquist
School of Meteorology
120 David L Boren Blvd., Suite 5900
Norman, OK 73072
ryan.lagerquist@ou.edu

David John Gagne II
National Center for Atmospheric Research
3090 Center Green Dr.
Boulder, CO 80301
dgagne@ucar.edu

ABSTRACT

We apply deep learning to high-impact weather prediction, specifically to the task of predicting tornadoes in the next hour based on current NEXRAD weather radar observations. In this paper, we analyze the results using multiple model interpretation approaches and demonstrate that model interpretation, especially when physically constrained and verified through statistical analysis, can be used to gain physical insights for atmospheric science.

1 INTRODUCTION AND MOTIVATION

Traditional machine learning (ML) techniques have a long history in meteorology and have been demonstrated to improve the prediction of multiple weather phenomena (e.g., Haupt et al., 2008; Trafalis et al., 2003; Williams et al., 2008; Gagne et al., 2009; Cintineo et al., 2014; Clark et al., 2015; McGovern et al., 2017). Many of these methods were chosen based on their intepretability, such as decision trees, which can be easily understood by domain scientists. Recently, deep learning (DL) has also proven to be a versatile and powerful tool for atmospheric science through improved prediction and understanding of convective hazards (McGovern et al., 2019; Gagne et al., 2019), estimation of sea-ice concentration (Wang et al., 2016), predicting tropical-cyclone intensity (Wimmers et al., 2019; Gagne et al., 2020), detecting extreme-weather patterns in model output (Racah et al., 2017; Kurth et al., 2018; Lagerquist et al., 2019a), and replacing parameterizations in physical models (Rasp et al., 2018; Brenowitz & Bretherton, 2018; 2019). However, deep learning is generally viewed as a "black box," especially by domain scientists. With an ultimate goal in mind of truly incorporating machine learning into a cycle of knowledge discovery with Earth scientists, we study the feasibility of using interpretation techniques for DL models to gain physical insights into the task of predicting tornadoes.

Interpretation techniques for DL are relatively recent and are just beginning to gain popularity in atmospheric science (e.g., Herman & Schumacher, 2018; McGovern et al., 2019; Toms et al., 2019b). In this paper, we focus specifically on saliency maps (Simonyan et al., 2014), backward optimization (Olah et al., 2017; 2018), and visualization of the most activated examples (Zeiler & Fergus, 2014). To help physically verify the potential scientific insights identified by the saliency maps, we perform "sanity checks" on the maps as proposed by Adebayo et al. (2018). These checks enable the domain scientist to verify that the knowledge identified by the deep learning methods are statistically significant. We also introduce a constraint function to ensure that backward optimization results remain physically plausible.

Our focus in this paper is on the task of tornado prediction. We train a Convolutional Neural Network (CNN, LeCun et al., 2015) to predict the probability that a storm will produce a tornado in the next hour given data similar to those used by meteorologists in real-time. We then appply saliency maps and backward optimization to understand the physical relationships learned by the CNN. This work is similar to the tornado work presented in (McGovern et al., 2019) but with several key differences. First, the model used here is trained with 3-D radar data, rather than 2-D. Second, we present a constraint loss function for backward optimization that encourages physical realism of the synthetic storms. Third, we use the sanity checks on the saliency maps to ensure that interpretation results are statistically significant. Finally, we visualize the examples that activate the most discriminative neurons with minimal cross-correlation to sample the variety of storm modes encoded by the CNN.

2 TORNADO PREDICTION

Due to space, we briefly describe the data and machine learning setup for tornado prediction. Complete details are given in Lagerquist et al. (2019b). Our goal was to use data that was as similar as possible to the data available to forecasters in real-time. Given the rarity of actual tornadoes, we train retrospectively. We use two datasets as input to the deep learning model: Gridded NEXRAD WSR-88D (GridRad) (Homeyer & Bowman, 2017) and the Rapid Refresh model (RAP) (Benjamin et al., 2016). Labels come from the National Weather Service (NWS) tornado reports database (NWS, 2016). GridRad contains merged radar data from all Weather Surveillance Radar 1988 Doppler (WSR-88D) (Crum & Alberty, 1993) sites in the continental United States (CONUS). Each radar scans a different part of the atmosphere, and where multiple radars scan the same point, they generally do so with differing resolution and errors. Merging data from all radars allows the data to be represented on a common Cartesian grid, and the merging algorithm includes quality-control measures that cannot be applied to single-radar data. This gives us a high-resolution 3D scan of the atmospheric available every 5 minutes across the CONUS. The RAP is a physical weather model that provides simulated environmental soundings with information on the wind and temperature from the surface to the upper atmosphere. We use RAP-simulated soundings because observed soundings are too sparse (CONUS has only 92 measurement sites, which launch soundings only once every 12 hours). The RAP produces consistent soundings on a grid across CONUS at a 1 hour interval.

To train the CNN, we use GridRad to create storm-centered radar images and the RAP to create a proximity sounding, representing the environment in which the storm will evolve over the next hour. Each input to the CNN (also called an "example") is one thunderstorm at one time. Our pre-processing methods (described in detail in Lagerquist et al. (2019b)) are used to identify and track individual storm cells as well as to link storms with labels from the NWS. The input image to the CNN uses 3D images from the GridRad radar data, encompassing each radar variable. The image is a 48 km$\times$ 48 km $\times$ 12-km equidistant grid with 1.5-km horizontal spacing (which makes input images $32 \times 32 \times 12$ pixels), 1.0-km vertical spacing, and storm motion pointing to the right. The grid is aligned with storm motion because tornadoes usually occur on the right-rear flank of the storm, regardless of its direction of motion. The proximity sounding is from the nearest RAP grid cell to point P, where P is the storm center extrapolated 30 minutes ahead (the median of the 0–1-hour prediction window) along the storm's motion vector.

The CNN used in this work had three layers of convolution and pooling, followed by two dense layers. The 3-D inputs are: radar reflectivity, which generally increases with storm strength and precipitation rate; spectrum width, which generally increases with mean wind speed and turbulence; vorticity, which is the rotational component of the wind; and divergence, which is the wind flux away from a point. Each of these is given to the CNN as an image. The environmental sounding is a 1-D input. The full architecture of the CNN is given in the Appendix in Figure 4.

The data is split into training, validation, and testing. Training data comes from the period 2012-2015. Validation comes from 2016-2018 and testing is from 2011, which was a year with high tornadic activity. Each period excludes the last week to ensure that temporal autocorrelation is eliminated across datasets. One example for the CNN is one thunderstorm at one time step. The training set contains 170 562 examples, 2.83% of which are tornadic; the validation set contains 85 056 examples (2.18% tornadic); and the testing set contains 158 781 examples (3.16% tornadic). Given that the focus of this paper is on the interpretation, we show the objective performance analysis in the Appendix in Figure 5. Area under the receiver-operating-characteristic curve (AUC; Metz, 1978) is $\sim$0.93, which surpasses the 0.9 threshold generally considered for excellent performance (Luna-Herrera et al., 2003; Muller et al., 2005; Mehdi et al., 2011).

3 MODEL INTERPRETATION METHODS

We briefly describe the three model interpretation methods that we used. We have applied additional methods but do not have the space to describe them here. Instead, we focus on saliency maps, backward optimization, and neuron visualizations and rankings.

3.1 SALIENCY MAPS

Saliency (Simonyan et al., 2014) is defined for each scalar predictor (*i.e.*, each variable at each grid point). The saliency of scalar predictor x is $\frac{\partial a}{\partial x}\Big|_{x=x_0}$, where a is the activation of a neuron in the model and x_0 is the value of x in a testing example. In this work we compute saliency for the output neuron, whose activation is the predicted tornado probability. Saliency can be computed for all scalar predictors, leading to a map that can be overlain with the input data to highlight predictors to which the most model is most sensitive and the direction (positive or negative) of this sensitivity.

Adebayo et al. (2018) proposed three sanity checks to ensure that saliency maps truly depend on relationships learned by the model, rather than artifacts caused by the model architecture or grid dimensions, such as Buell patterns in principal-component analysis (Buell, 1979). The first check is the edge-detector test, which compares the saliency maps produced by the trained model with maps produced by an untrained edge-detector. The second is the model-parameter-randomization test, which compares saliency maps from the trained model before and after randomizing the weights in some layers. The third is the data-randomization test, which compares saliency maps from the model trained with the true data to maps produced by a model trained on random labels. However, we were unable to create a model that overfits random labels, so we present results only for the first two checks. Each sanity check produces a set of dummy saliency maps (one for each testing example), which we compare to the actual saliency maps. To assess statistical significance, we apply a two-tailed Monte Carlo test (Dwass, 1957) to the composite difference (Appendix B).

3.2 BACKWARD OPTIMIZATION (BWO)

BWO (Erhan et al., 2009) creates a synthetic input that extremizes (minimizes or maximizes) the activation of a particular neuron in the model. BWO is sometimes called activation maximization (Erhan et al., 2009), feature optimization (Olah et al., 2017), or optimal input (Toms et al., 2019a). The BWO procedure is basically training in reverse. During training, BWO is used to adjust weights in a way that minimizes the loss function. During BWO, gradient descent is used to adjust predictor values in a way that extremizes the neuron activation. As for saliency maps, we focus on the output neuron, whose activation is predicted tornado probability.

At each iteration of gradient descent for BWO, the synthetic example is updated via the following rule. $\mathbf{X}$ is the tensor of predictor values; J is the loss function; $\frac{\partial J}{\partial \mathbf{X}}$ is a gradient tensor with the same dimensions as $\mathbf{X}$; and α is the learning rate, usually a positive number $\ll 1$. Both α and the number of iterations are hyperparameters.

$$\mathbf{X} \leftarrow \mathbf{X} - \alpha \frac{\partial J}{\partial \mathbf{X}} \tag{1}$$

In the simplest framework, $J = (p - p^*)^2$, where p is the CNN-generated class probability and p^* is the desired probability (0.0 or 1.0). Thus, Equation 1 can be written as follows.

$$\mathbf{X} \leftarrow \mathbf{X} - 2\alpha(p - p^*) \frac{\partial p}{\partial \mathbf{X}} \tag{2}$$

Gradient descent requires a starting point or "initial seed". Some options are all-zeros, random noise, or an actual testing example. The advantage of all-zeros and random is that the initial seed does not resemble a real example, so the synthetic example ultimately produced is more novel. The disadvantage is that, because the initial seed is very unrealistic, the synthetic example can be very unrealistic as well. Starting from an actual example encourages more realistic output. However, this is not guaranteed, so we add constraints to the loss function (Equation 3) to encourage physical realism of the synthetic example.

$$\begin{aligned} J = (p - p^*)^2 + \lambda_2 \|\mathbf{X} - \mathbf{X_0}\|^2 + \lambda_{\text{minmax}} \sum_j \|\max(\mathbf{X_j^{\text{min}}} - \mathbf{X_j}, \mathbf{0})\|^2 \\ + \lambda_{\text{minmax}} \sum_j \|\max(\mathbf{X_j} - \mathbf{X_j^{\text{max}}}, \mathbf{0})\|^2 \end{aligned} \tag{3}$$

The first term is the unconstrained loss function; the second term is the L_2 penalty on the difference between the original example $\mathbf{X_0}$ and synthetic example $\mathbf{X}$; the third term is the penalty for violating minimum-constraints; and the fourth term is the penalty for violating maximum-constraints. $\mathbf{X_j}$ is a tensor with values of the j^{th} variable only; $\mathbf{X_j^{min}}$ is a tensor of the same shape, where every value is the minimum allowed for the j^{th} variable; $\mathbf{X_j^{max}}$ is a tensor of the same shape, where every value is the maximum allowed for the j^{th} variable; and $\mathbf{0}$ is the tensor of the same shape, where every value is zero. For instance, if the j^{th} variable is fractional relative humidity (where 0 is 0% and 1 is 100%), $\mathbf{X_j^{min}} = \mathbf{0}$ and $\mathbf{X_j^{max}} = \mathbf{1}$. λ_2 is the strength of the L_2 penalty, and λ_{minmax} is the strength of the min-max penalty, both hyperparameters. The constraints are given in Appendix B.

3.3 Neuron Ranking by Area Under the ROC Curve

Each neuron in a convolutional neural network is optimized to make the predicted classes more separable from each other. Due to randomness in the stochastic gradient descent process, not all neurons are equally good at separating different classes. However, we can rank the quality of a set of given neurons by evaluating the discrimination ability of their output with the Area Under the ROC Curve (AUC) metric. Although AUC is generally calculated for probabilistic predictions, the approach can be applied to any continuous range of values that has binary labels associated with it. For binary classification problems, AUC values well above 0.5 indicate that increasing the activation values increases the probability of the positive class, but AUC values well below 0.5 indicate that increasing the activation values decreases the probability of the positive class. After ranking the neurons by AUC, we calculate the Pearson correlation matrix among all neuron activations. Since there is strong correlation among some of the neurons, we sequentially select the top neurons in terms of AUC that also have a maximum correlation with any previously selected neuron below a threshold of 0.5. Since the storms are already rotated in the direction of storm motion, we composite radar reflectivity of the top 30 examples to reveal the preferred storm mode for each neuron.

4 Results

Figure 1 shows composite saliency maps for the 100 best hits, defined as the 100 positive examples (storms that are tornadic in the next hour) with the highest predicted probabilities from the CNN (average of 99.2%). The saliency map is computed independently for each example, and the results are composited via probability-matched means (PMM; Ebert, 2001), which preserves spatial structure better than taking the mean for each scalar predictor independently. Figure 1a shows the actual saliency map. Saliency is maximized on the right-rear flank of the storm, where the reflectivity core and mesocyclone intersect and a potential tornado would be expected (Klemp & Rotunno, 1983). Tornado probability increases with reflectivity, vorticity, and spectrum width at all heights. Saliency for reflectivity and spectrum width is highest at 10 km above ground level (AGL), while saliency for vorticity is highest at 2 km AGL. Thus, the model is particularly sensitive to low-level rotation and storm depth. Figures 1b-c show dummy saliency maps, created by two of the sanity checks discussed in Section 3.1. Stippling shows where differences between rescaled values (Appendix B) in the dummy and actual saliency maps are significant. According to the edge-detector test (Figure 1b), 20% of values are significantly different, primarily at 2 km AGL. Above 2 km, the shape of the saliency map (rescaled values) can be mostly replicated by an untrained edge-detection filter, but raw values cannot (note the difference between color scales). According to the model-parameter-randomization test (Figure 1c), 23% of values are significantly different, which suggests that actual saliency maps truly depend on learned weights in the given layer. Results for other layers, both convolutional and dense, are similar. In general, saliency maps for the GridRad model cannot be trivially replicated, which suggests that they truly reflect physical relationships learned by the model.

Figure 2 shows synthetic storms created by applying BWO to the best hits, with the goal of decreasing tornado probability to 0.0. The results are composited via PMM. On average, the adjustments made by BWO decrease tornado probability from 99.2% to 6.9%. In general, BWO decreases all three radar variables in the core and mesocyclone, making the storms weaker, and increases all three variables in the surrounding area, making the storms less discrete (isolated from surrounding convection). The effect of physical constraints is most obvious in the synthetic soundings (Figure 2e-f). Unconstrained BWO creates sharp discontinuities in both the dewpoint and temperature profiles that are not realistic.

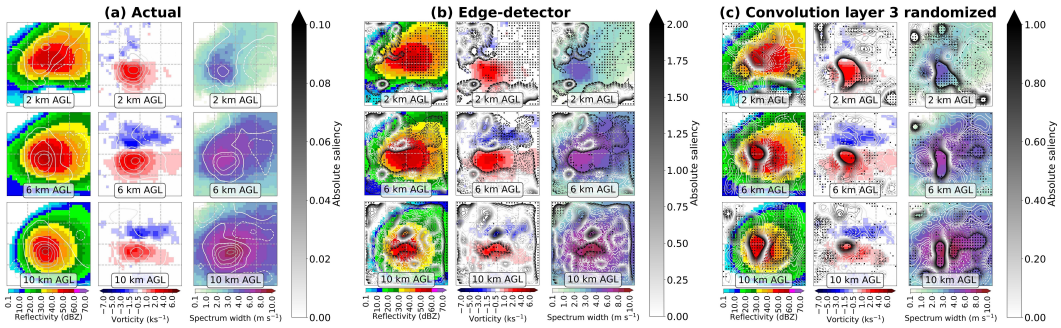

Figure 1: Sanity checks on saliency maps for the 100 best hits (composited via PMM). Storm motion points to the right. Heat maps show the radar fields (predictors), while line contours show saliency. Solid contours indicate positive saliency (tornado probability increases with the underlying radar value), while dashed contours indicate negative saliency (tornado probability decreases with the underlying radar value). [a] Actual saliency map. [b] Saliency map produced by an untrained edge-detector. [c] Saliency map produced after randomizing weights in the third convolutional layer. In panels b-c, stippling shows where the actual and dummy saliency maps are significantly different at the 95% confidence level.

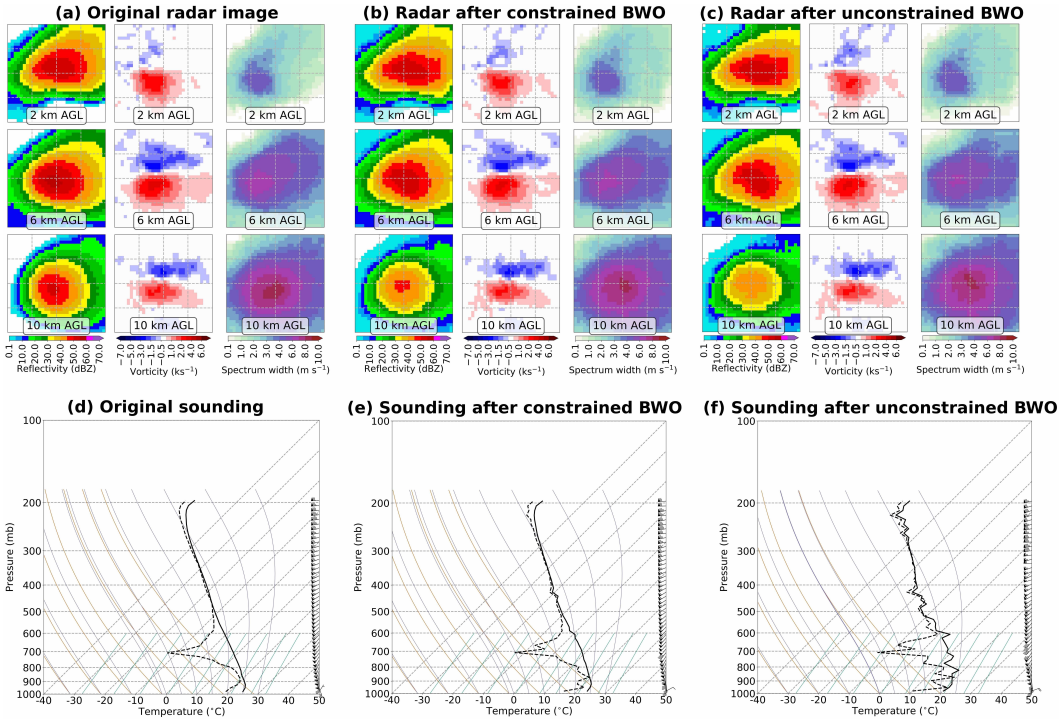

Figure 2: Backward optimization for the 100 best hits (composited via PMM). Storm motion points to the right. [a] Original radar image, before BWO. [b] Synthetic radar image, created by BWO with constraints. [c] Same but for unconstrained BWO. [d-f] Same as a-c but for the proximity sounding, plotted as a skew-T log-p diagram, a common tool for visualizing soundings. The thick solid black line is air temperature; the thick dashed black line is dewpoint temperature; and vectors along the right-hand side are wind barbs. The y-axis is pressure, which decreases with height, so these plots show vertical profiles.

The composites of the storms that activate the most discriminative and minimally-correlated neurons are shown in Fig. 3. The top neuron corresponds to a supercellular storm mode and the composite of the 100 best hits from Fig. 1. The other top neurons highlight either other supercellular modes

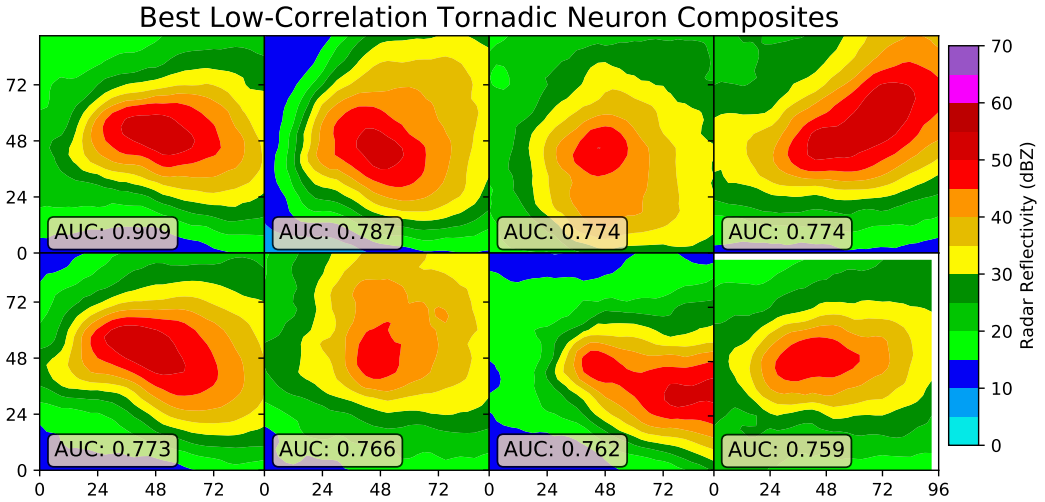

Figure 3: Composites of the top 30 storms that activate the top minimally-correlated neurons in the last layer of the convolutional neural network along with the associated AUC for that neuron.

where the storm has more of an across-track extent or is in a more linear storm mode. The more linear modes tend to have a lower AUC, which matches with the lower skill human forecasters have in predicting tornadoes from these storm modes. The best non-tornadic neurons focus on storms with weak radar reflectivity and small single-cell modes.

5 Discussion and Future Work

We have presented the results of applying multiple model interpretation models to the task of predicting tornadoes in a storm over the next hour. The model interpretation methods identify knowledge that is consistent with current knowledge of tornadogenesis. We demonstrated that the sanity checks for saliency maps can be used to verify the value of the knowledge. This can help to prevent confirmation bias when examining model interpretation results.

We also demonstrated that adding physically based constraints to the loss function for BWO can improve the physical realism of synthetic storms. While the constraints are not sufficient to fully solve the need for physically based results, they are promising and we are continuing develop approaches to physically constraint the models and the interpretation results.

We finally demonstrated that CNNs can encode a variety of storm modes in their internal neurons, and that the discriminative skill of each neuron matches subjectively with the skill or lack thereof that human forecasters have in predicting tornadoes from these modes. Further analysis with saliency maps and other interpretation techniques will reveal the areas of interest to the neural network for each storm mode along with differences in 3D structure that may not be apparent from the current 2D visualization.

Acknowledgments

The authors thank Cameron Homeyer for the GridRad data. This material is based upon work supported by the National Science Foundation under Grant No. EAGER AGS 1802627 and the NCAR Cooperative Agreement AGS-1852977. Funding was also provided by NOAA/Office of Oceanic and Atmospheric Research under NOAA-University of Oklahoma Cooperative Agreement #NA16OAR4320115, U.S. Department of Commerce. Computing for this project was performed at the OU Supercomputing Center for Education & Research (OSCER) at the University of Oklahoma (OU) and the NCAR Casper data analysis and visualization cluster (Computational and Information Systems Laboratory, 2020).

REFERENCES

J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. In *Conference on Neural Information Processing Systems*, Montréal, Canada, 2018. Neural Information Processing Systems Foundation.

S.G. Benjamin, S.S. Weygandt, J.M. Brown, M. Hu, C.R. Alexander, T.G. Smirnova, J.B. Olson, E.P. James, D.C. Dowell, G.A. Grell, H. Lin, S.E. Peckham, T.L. Smith, W.R. Moninger, J.S. Kenyon, and G.S. Manikin. A North American hourly assimilation and model forecast cycle: The Rapid Refresh. *Monthly Weather Review*, 144(4):1669–1694, 2016.

N.D. Brenowitz and C.S. Bretherton. Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, 45(12):6289–6298, 2018.

N.D. Brenowitz and C.S. Bretherton. Spatially extended tests of a neural network parametrization trained by coarse-graining. *Journal of Advances in Modeling Earth Systems*, 11, 2019.

C.E. Buell. On the physical interpretation of empirical orthogonal functions. In *Conference on Probability and Statistics*, Banff, Canada, 1979. American Meteorological Society.

F. Chollet. *Deep Learning with Python*. Manning, Shelter Island, New York, 2018.

J L Cintineo, M J Pavolonis, J M Sieglaff, and D T Lindsey. An empirical model for assessing the severe weather potential of developing convection. *Weather and Forecasting*, 29(3):639–653, 2014.

A.J. Clark, A. MacKenzie, A. McGovern, V. Lakshmanan, and R.A. Brown. An automated, multiparameter dryline identification algorithm. *Weather and Forecasting*, 30(6):1781–1794, 2015.

Computational and Information Systems Laboratory. Cheyenne: HPE/SGI ICE XA System (NCAR Community Computing). Technical report, National Center for Atmospheric Research, 2020. URL https://doi.org/10.5065/d6rx99hx.

T.D. Crum and R.L. Alberty. The WSR-88D and the WSR-88D operational support facility. *Bulletin of the American Meteorological Society*, 74(9):1669–1687, 1993.

M. Dwass. Modified randomization tests for nonparametric hypotheses. *The Annals of Mathematical Statistics*, 28(1):181–187, 1957.

E.E. Ebert. Ability of a poor man's ensemble to predict the probability and distribution of precipitation. *Monthly Weather Review*, 129(10):2461–2480, 2001.

D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. Technical report, 2009.

D. J. Gagne, C. Rozoff, and J. Vigh. Probabilistic rapid intensification prediction with convolutional neural networks and hwrf. In *Proceedings of the 19th Conference on Artificial Intelligence for Environmental Science*, pp. J43.2, Boston, MA, 2020. American Meteorological Society. URL https://ams.confex.com/ams/2020Annual/meetingapp.cgi/Paper/367819.

D.J. Gagne, A. McGovern, and J. Brotzge. Classification of convective areas using decision trees. *Journal of Atmospheric and Oceanic Technology*, 26(7):1341–1353, 2009.

D.J. Gagne, S.E. Haupt, D.W. Nychka, and G. Thompson. Interpretable deep learning for spatial analysis of severe hailstorms. *Monthly Weather Review*, 147(8):2827–2845, 2019.

S. Haupt, A. Pasini, and C. Marzban (eds.). *Artificial Intelligence Methods in the Environmental Sciences*. Springer, 2008.

G.R. Herman and R.S. Schumacher. Dendrology in numerical weather prediction: What random forests and logistic regression tell us about forecasting. *Mon. Wea. Rev.*, 146:1785–1812, 2018. URL https://doi.org/10.1175/MWR-D-17-0307.1.

G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv e-prints*, 1207(0580), 2012.

A. Hoerl and R. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

A. Hoerl and R. Kennard. Ridge Regression. In S. Kotz (ed.), *Encyclopedia of Statistical Sciences*, volume 8. Wiley, 1988.

C.R. Homeyer and K.P. Bowman. Algorithm Description Document for Version 3.1 of the Three-Dimensional Gridded NEXRAD WSR-88D Radar (GridRad) Dataset. Technical report, University of Oklahoma, 2017. URL `http://gridrad.org/pdf/GridRad-v3.1-Algorithm-Description.pdf`.

David D. Jensen and Paul R. Cohen. Multiple comparisons in induction algorithms. *Machine Learning*, 38(3):309–338, 2000.

J.B. Klemp and R. Rotunno. A study of the tornadic region within a supercell thunderstorm. *Journal of the Atmospheric Sciences*, 40(2):359–377, 1983.

T. Kurth, S. Treichler, J. Romero, M. Mudigonda, N. Luehr, E. Phillips, A. Mahesh, M. Matheson, J. Deslippe, M. Fatica, Prabhat, and M. Houston. Exascale deep learning for climate analytics. In *International Conference for High Performance Computing, Networking, Storage, and Analysis*, Dallas, Texas, 2018. Institute of Electrical and Electronics Engineers (IEEE).

R. Lagerquist, A. McGovern, and D.J. Gagne. Deep learning for spatially explicit prediction of synoptic-scale fronts. *Weather and Forecasting*, 34(4):1137–1160, 2019a.

R. Lagerquist, A. McGovern, C. Homeyer, D.J. Gagne, and T. Smith. Deep learning on three-dimensional multiscale data for next-hour tornado prediction. *Monthly Weather Review*, conditionally accepted, 2019b.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

J. Luna-Herrera, G. Martinez-Cabrera, R. Parra-Maldonado, J.A. Enciso-Moreno, J. Torres-Lopez, F. Quesada-Pascual, R. Delgadillo-Polanco, and S.G. Franzblau. Use of receiver operating characteristic curves to assess the performance of a microdilution assay for determination of drug susceptibility of clinical isolates of Mycobacterium tuberculosis. *European Journal of Clinical Microbiology and Infectious Diseases*, 22(1):21–27, 2003.

A.L. Maas, A.Y. Hannun, and A.Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning*, Atlanta, Georgia, 2013. International Machine Learning Society.

A. McGovern, K.L. Elmore, D.J. Gagne, S.E. Haupt, C.D. Karstens, R. Lagerquist, T. Smith, and J.K. Williams. Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bulletin of the American Meteorological Society*, 98(10):2073–2090, 2017.

A. McGovern, R. Lagerquist, D.J. Gagne, G.E. Jergensen, K.L. Elmore, C.R. Homeyer, and T. Smith. Making the black box more transparent: Understanding the physical implications of machine learning. *Bulletin of the American Meteorological Society*, 100(11):2175–2199, 2019.

T. Mehdi, N. Bashardoost, and M. Ahmadi. Kernel smoothing for ROC curve and estimation for thyroid stimulating hormone. *International Journal of Public Health Research*, Special Issue: 239–242, 2011.

C.E. Metz. Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, 8(4):283–298, 1978.

M.P. Muller, G. Tomlinson, T.J. Marrie, P. Tang, A. McGeer, D.E. Low, A.S. Detsky, and W.L. Gold. Can routine laboratory tests discriminate between severe acute respiratory syndrome and other causes of community-acquired pneumonia? *Clinical Infectious Diseases*, 40(8):1079–1086, 2005.

NWS. *Storm Data* preparation. National Weather Service Instruction 10-1605. Technical report, 2016.

C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. URL https://distill.pub/2017/feature-visualization.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization: How neural networks build up their understanding of images. *Distill*, 2018.

E. Racah, C. Beckham, T. Maharaj, S.E. Kahou, 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*, Long Beach, California, 2017. Neural Information Processing Systems.

S. Rasp, M.S. Pritchard, and P. Gentine. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39):9684–9689, 2018.

P.J. Roebber. Visualizing multiple measures of forecast quality. *Weather and Forecasting*, 24(2): 601–608, 2009.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualizing image classification models and saliency maps. *arXiv pre-prints*, 1312, 2014.

B.A. Toms, E.A. Barnes, and I. Ebert-Uphoff. Physically interpretable neural networks for the geosciences: Applications to Earth system variability. *arXiv pre-prints*, 1912(01752), 2019a.

Benjamin A Toms, Elizabeth A Barnes, and Imme Ebert-Uphoff. Physically Interpretable Neural Networks for the Geosciences: Applications to Earth System Variability. *arXiv preprint arXiv:1912.01752*, December 2019b. URL https://arxiv.org/abs/1912.01752.

Theodore B. Trafalis, Huseyin Ince, and Michael B. Richman. Tornado detection with support vector machines. In *Computational Science – ICCS 2003*, volume 2660/2003 of *Lecture Notes in Computer Science*, pp. 289–298. Springer Berlin / Heidelberg, 2003.

L. Wang, K.A. Scott, L. Xu, and D.A. Clausi. Sea ice concentration estimation during melt from dual-pol SAR scenes using deep convolutional neural networks: A case study. *IEEE Transactions on Geoscience and Remote Sensing*, 54(8):4524–4533, 2016.

D.S. Wilks. "The stippling shows statistically significant grid points": How research results are routinely overstated and overinterpreted, and what to do about it. *Bulletin of the American Meteorological Society*, 97(12):2263–2273, 2016.

J.K. Williams, D. Ahijevych, S. Dettling, and M. Steiner. Combining observations and model data for short-term storm forecasting. In *Remote Sensing Applications for Aviation Weather Hazard Detection and Decision Support*, San Diego, CA, 2008. International Society for Optics and Photonics.

A. Wimmers, C. Velden, and J.H. Cossuth. Using deep learning to estimate tropical cyclone intensity from satellite passive microwave imagery. *Monthly Weather Review*, 147(6):2261–2282, 2019.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

A APPENDIX: MACHINE LEARNING DETAILS

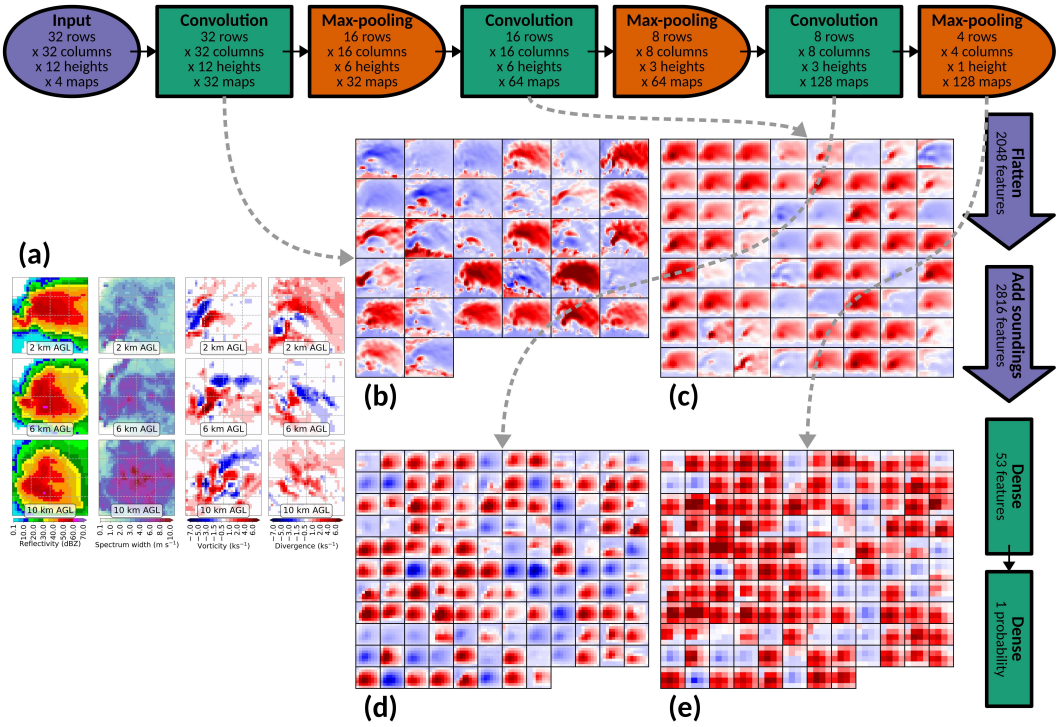

Figure 4: Architecture of the tornado-prediction CNN. The input is a $32 \times 32 \times 12$ grid with four variables. One example input at three of the twelve heights, is shown in panel a. The three convolution layers transform these variables into a successively increasing number of feature maps (lowest height level shown in panels b-e). The pooling layers successively halve spatial resolution. The sounding input is not shown but it follows the same architecture, with three convolution layers and three pooling layers. Maps from the two feature-detectors are flattened, concatenated together, and passed through two dense layers, terminating with one scalar output (next-hour tornado probability).

The architecture of the CNN used is shown in Figure 4. The activation function for all convolution layers and the first dense layer is the leaky rectified linear unit (ReLU) (Maas et al., 2013) with a slope, $\alpha = 0.2$. The final dense layer uses the sigmoid activation function, which forces the output to range from $[0, 1]$, allowing it to be interpreted as a probability.

The CNN is trained with three types of regularization: L_2 regularization (Hoerl & Kennard, 1970; 1988) in the convolution layers, dropout (Hinton et al., 2012) in the first dense layer, and data augmentation (Section 5.2.5 of Chollet 2018) (Chollet, 2018). For the data augmentation, we apply the following small perturbations to each radar image during training: horizontal translation by 3 grid cells to the north, northeast, east, southeast, south, southwest, west, and northwest; rotation in the horizontal plane by -15°, +15°, -30°, and +30°; and five additions of Gaussian noise with a standard deviation of 0.1. Inside the CNN, all predictors are normalized to z-scores (with a mean of 0.0 and standard deviation of 1.0), so Gaussian noise has an equal impact on all predictors. The perturbations are applied separately, turning each training example into 18 (the original example plus 17 perturbed ones). The specific perturbations were determined by a hyperparameter experiment, as were the L_2 strength (0.001), dropout rate (50%), and number of dense layers (two, as shown in Figure 4). The objective of the hyperparameter experiment was to maximize AUC on the validation data.

CNN objective performance on the testing data is shown in Figure 5. We measured performance using AUC ($\sim$0.93) and with a performance diagram (Roebber, 2009). AUC is measured from the probability of detection (POD), which is the fraction of tornadic examples that are correctly forecast and the probability of false detection (POFD), which is the fraction of non-tornadic examples that are incorrectly forecast. Performance diagrams plot the success ratio, which is the fraction of tornadic

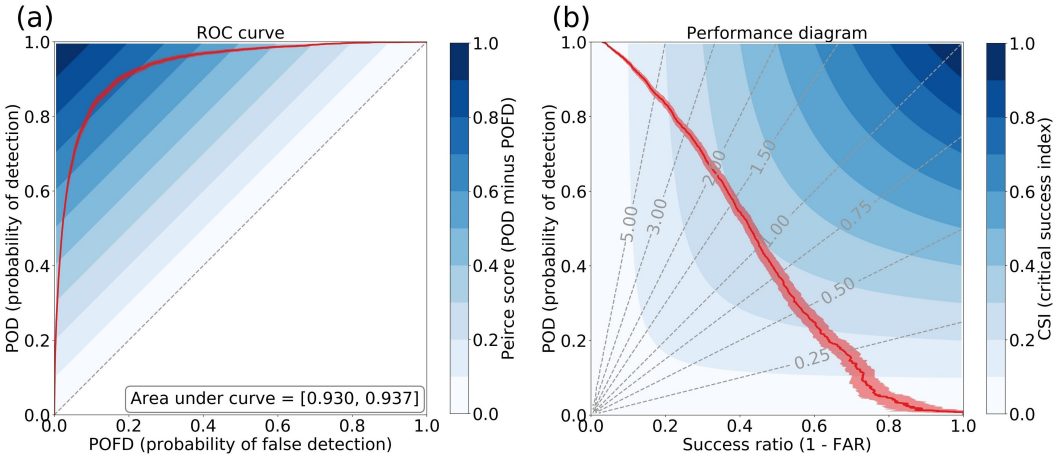

Figure 5: Performance of the tornado prediction CNN on testing data. Dark lines show the mean, and light shading shows the 95% confidence interval, determined by bootstrapping the testing examples 1000 times. Dashed grey lines in the performance diagram show frequency bias, which is the ratio of forecast to actual tornadoes and should ideally be 1.0.

forecasts that are correct against POD. The ROC curve is insensitive to event frequency while the performance diagram is highly sensitive to event frequency. For rare events such as tornadoes, it is difficult to achieve a high probability of detection with low false-alarm ratio, so the curve tends to be near the bottom-left. Nonetheless, the high AUC suggests that our model performs well enough that it is worth querying via interpretation methods.

B Appendix: Method details

B.1 Saliency Map Statistical Checking

The two-tailed Monte Carlo test (Dwass, 1957) on composite differences is described below.

1. Rescale all saliency values to percentiles to ensure values between actual and dummy saliency maps can be compared.
2. For each of 10 000 iterations:
   (a) Randomly shuffle examples between the actual (A) and dummy sets (D), yielding A' and D'. Both A' and D' contain a mix of actual and dummy saliency maps.
   (b) Take the probability-matched mean (PMM; Ebert, 2001) for both A' and D'. Record the randomized composite difference (PMM from A' minus PMM from D') for each predictor.
3. For each scalar predictor:
   (a) Find the true composite difference (PMM over set A minus PMM over set D).
   (b) Find the percentile of the true composite difference in the distribution of 20 000 randomized composite differences.
   (c) If the percentile is < 2.5 or > 97.5, the difference is significant at the 95% confidence level.

Step 2a shuffles entire examples, rather than shuffling independently for each grid point or each scalar predictor. This preserves spatial and cross-channel correlations in the data, which obviates the need to explicitly control the false-discovery rate for multiple comparisons (Jensen & Cohen, 2000; Wilks, 2016).

B.2 BWO details

In this work we set $\lambda_2 = 1$ and $\lambda_{\text{minmax}} = 10$, with the following constraints.

- Minimum reflectivity in radar image = 0 dBZ

- Minimum spectrum width in radar image = 0 m s^{-1}
- Minimum θ_v in sounding = 0 K
- Minimum relative humidity in sounding = 0%
- Minimum specific humidity in sounding = 0 g kg^{-1}
- Maximum relative humidity in sounding = 100%
- Maximum specific humidity in sounding = 1000 g kg^{-1}