MAKING THE BLACK BOX MORE TRANSPARENT

Understanding the Physical Implications of Machine Learning

Amy McGovern, Ryan Lagerquist, David John Gagne II, G. Eli Jergensen, Kimberly L. Elmore, Cameron R. Homeyer, and Travis Smith

Machine learning model interpretation and visualization focusing on meteorological domains are introduced and analyzed.

Machine learning (ML) and deep learning (DL; LeCun et al. 2015) have recently achieved breakthroughs across a variety of fields, including the world's best Go player (Silver et al. 2016, 2017), medical diagnosis (Rakhlin et al. 2018), and galaxy classification (Dieleman et al. 2015). Simple forms of ML (e.g., linear regression) have been used in meteorology since at least the 1950s (Malone 1955), and ML has been used extensively to forecast convective hazards since the mid-1990s. Kitzmiller et al. (1995) use linear regression to forecast the probability of tornadoes, large hail, or damaging wind; Billet et al. (1997) use linear regression to forecast hail probability and size; Marzban and Stumpf (1996, 1998) use neural networks to forecast the probability of tornadoes and damaging wind, respectively; and Marzban and Witt (2001) use neural networks to forecast hail size. Gagne et al. (2013, 2017a) use random forests to forecast hail probability at 1-day lead time; McGovern et al. (2014) and Williams (2014) use random forests to forecast convectively induced aircraft turbulence; while Cintineo et al. (2014, 2018) use naïve Bayes to forecast the probability of tornadoes, large hail, and damaging wind. DL is also beginning to be used in meteorology, with applications including hail prediction (Gagne et al. 2019) and detection of extreme weather patterns such as tropical cyclones, atmospheric rivers, and synoptic-scale fronts (Liu et al. 2016; Mahesh et al. 2018; Kunkel et al. 2018; Lagerquist et al. 2019b). The authors have extensive

AFFILIATIONS: McGovern and Jergensen—University of Oklahoma, Norman, Oklahoma; Lagerquist—Cooperative Institute for Mesoscale Meteorological Studies, and University of Oklahoma, Norman, Oklahoma; Gagne—National Center for Atmospheric Research, Boulder, Colorado; Elmore and Smith—Cooperative Institute for Mesoscale Meteorological Studies, University of Oklahoma, and NOAA/National Severe Storms Laboratory, Norman, Oklahoma; Homeyer—School of Meteorology, University of Oklahoma, Norman, Oklahoma
CORRESPONDING AUTHOR: Amy McGovern, amcgovern@ou.edu

The abstract for this article can be found in this issue, following the table of contents.
DOI:10.1175/BAMS-D-18-0195.1

A supplement to this article is available online (10.1175/BAMS-D-18-0195.2)

In final form 20 June 2019

experience using ML to improve forecasting and understanding of weather phenomena (Gagne et al. 2017a,b; Lagerquist et al. 2017; McGovern et al. 2017; Gagne et al. 2019; Lagerquist et al. 2018). Many of these products have been used by human meteorologists in experiments and day-to-day operations.

Despite its wide adoption in meteorology, ML is often criticized by forecasters and other end users as being a "black box" because of the perceived inability to understand how ML makes its predictions. This phenomenon is not exclusive to meteorology, and many ML practitioners and users have recently begun to focus on this interpretability problem (Olah et al. 2017; Lipton 2016; NeurIPS Foundation 2018; Molnar 2018).

The main contribution of this paper is to synthesize and analyze multiple approaches to model interpretation and visualization (MIV) for meteorology. MIV is useful in all phases of ML development (Selvaraju et al. 2017). Initially, MIV can be used to aid with debugging, enabling the domain scientists and data scientists to ensure that the models are focusing on the most physically relevant aspects of the problem. During deployment, MIV can be used to help the users gain trust in the model and to identify ideal scenarios as well as shortcomings. If ML surpasses human predictions, interpretation methods could be used to improve the humans' skills (Silver et al. 2016; Johns et al. 2015) and MIV could be used to identify new scientific hypotheses.

To ensure general results, we use both traditional ML and DL for meteorological phenomena at different spatiotemporal scales. At the synoptic scale, we apply DL to predict the probability of severe hail 24–48 h in advance across the continental United States (CONUS). At the mesoscale, we apply traditional ML to soundings from a mesoscale numerical model to predict winter precipitation type. At the storm scale, we use DL to predict the probability that a storm will produce a tornado within the next hour and traditional ML methods to classify a storm's convective mode. Lagerquist et al. (2018, 2019a), Gagne et al. (2019), McGovern et al. (2018), and Jergensen et al. (2019) focus on the training and evaluation of these models, while we focus on MIV.

MACHINE LEARNING.

We briefly review ML as needed for the MIV explanations, giving more detail on DL since it is newer to meteorology. McGovern et al. (2017) provides a more in-depth review of traditional ML methods for meteorology.

Decision trees. Decision trees, which can be understood as a flowchart where the decision points have been automatically learned by a computer, have been used in meteorology since the 1960s (e.g., Chisholm et al. 1968). Decision trees were built subjectively by human experts until the 1980s, when an objective learning algorithm was developed (Quinlan 1986). Their human readability has helped contribute to their popularity in many scientific domains. During training, at each branch node, the algorithm considers a number of potential questions that can split the data (e.g., is dewpoint ≥ 60°F?). Splits are chosen to minimize error on the predictions, which can be classifications, probabilities (Provost and Domingos 2003), or real valued (Breiman 1984).

Decision trees are brittle, meaning that small changes in the data can cause large changes in the final model. Ensemble approaches, such as random forests (RF; Breiman 2001) and gradient-boosted regression trees (GBRT; Friedman 2002), mitigate this problem by training ensembles of trees but this minimizes human readability. In RF, diversity is maintained by training each tree with a different subset of examples. In GBRT, the kth tree is fit to the error of the first $k - 1$ trees, rather than being fit to the target value. RF and GBRT are used successfully in meteorology (Williams et al. 2008a,b; Gagne et al. 2009; McGovern et al. 2014; Williams 2014; McGovern et al. 2015; Clark et al. 2015; Elmore and Grams 2016; Lagerquist et al. 2017).

Support-vector machines. Support-vector machines (SVM; Vapnik 1963) find a hyperplane in predictor space that can be used to linearly separate the data. SVMs can also be used for nonbinary classification (Franc and Hlavac 2002) or regression (Drucker et al. 1997). SVMs often use a kernel to transform the predictor space into another space where the problem is more easily separated. One such kernel, the radial basis function (RBF; Schölkopf et al. 1997), uses a Gaussian to transform the space. Linear SVMs, like decision trees, are human readable when there are few predictors, but high-dimensional linear SVMs and nonlinear SVMs are not easily interpreted by a human. Recent SVM applications in meteorology include Radhika and Shashi (2009), Rao et al. (2012), and Rajasekhar and Rajinikanth (2014).

Deep learning. DL is a subset of ML that specializes in leveraging spatiotemporal structure in the input data. DL is well suited for meteorology, because it can be applied directly to spatiotemporal grids and identify salient features at different spatiotemporal scales. The DL models we use here are convolutional neural networks (CNN; Fukushima and Miyake

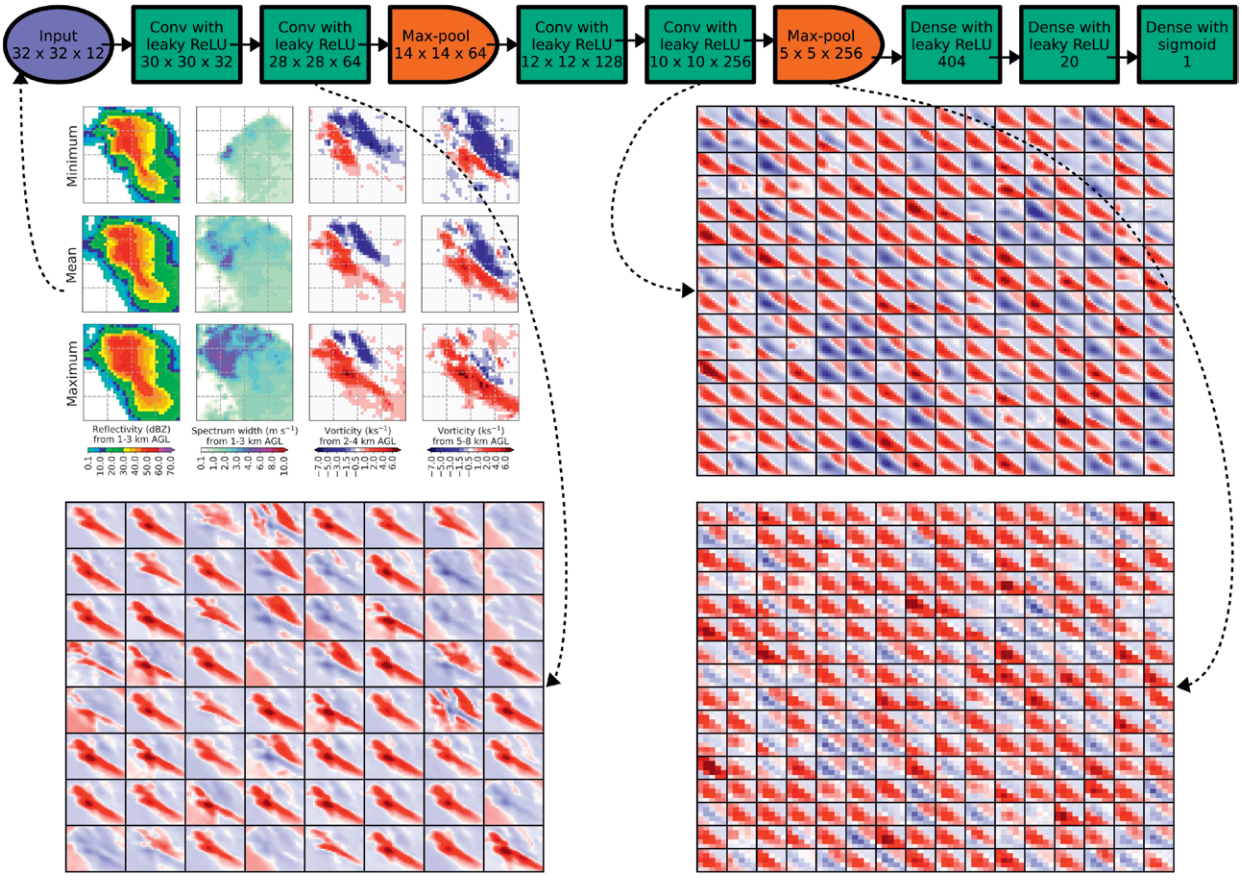

FIG. 1. Architecture of CNN used for tornado prediction. Each input example is a 32 × 32 grid of 12 different radar variables. In the feature maps produced by convolution and pooling layers, negative values are in blue and positive values are in red. (The input also includes five sounding variables, listed in Fig. 3, but these are omitted here for the sake of simplicity.) The first convolution layer transforms the 12 variables into 32 filters and removes one pixel around the edge. The first pooling layer downsamples feature maps to half resolution, thus halving the spatial dimensions. Other convolutional and pooling layers perform similar operations. Feature maps from the last pooling layer are flattened into a length-6,400 vector (5 × 5 × 256 = 6,400), which is transformed by the three dense layers into vectors of length 404, then 20, and then 1. The sigmoid activation function of the final dense layer forces the output (tornadogenesis probability) to the range [0, 1].

1982). The main components of a CNN are convolutional, pooling, and dense layers (Fig. 1). Each convolutional layer passes many convolutional filters over the input maps, where an input map consists of multiple "channels" (e.g., red, green, and blue channels for an image), creating one output map for each filter. The output maps (feature maps) are passed through a nonlinear activation function, such as the rectified linear unit (ReLU; Nair and Hinton 2010). The activation function must be nonlinear; otherwise, the net can learn only linear relationships. Supplemental Fig. ES1 is an animation of one filter.

During training, weights in the convolutional filters are updated via stochastic gradient descent (SGD; section 4.4.3 of Mitchell 1997) to minimize the loss function. The typical loss function for classification, adopted in this work, is cross entropy, defined as

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\log_2\left(\hat{y}_{ik}\right),$$

where N is the number of examples, K is the number of classes, y_{ik} is the true value (1 if the ith example belongs to the kth class, 0 otherwise), and $\hat{y}_{ik}$ is the predicted probability that the ith example belongs to the kth class. Cross entropy varies from [0, ∞), and the optimal score is zero.

Pooling layers reduce the resolution of feature maps, which allows deeper convolutional layers (farther to the right in Fig. 1) to learn larger-scale features along with some amount of translational invariance. Deeper layers learn higher-level abstractions, because the feature maps are larger in scale and have passed

through more nonlinear transformations (section 5.2.1 of Chollet 2018). Supplemental Fig. ES2 is an animation of maximum pooling.

Most CNNs end with one or more dense layers (section 3.1.1 of Chollet 2018; our Fig. 1), which are identical to hidden layers in a traditional neural net (Haykin 2001). Weights in the dense layers are learned via SGD, simultaneously with those in the convolutional layers. For binary classification, the final prediction is one value (probability of event). For K-class problems ($K > 2$), the final prediction consists of K values (one probability for each class).

For traditional ML methods presented in this work, we use scikit-learn (Pedregosa et al. 2011). For DL, we use Keras (Chollet 2015). All models are trained on the training data; hyperparameters (parameters not adjusted during training) are selected by which model performs best on the validation data; and final results are reported on the testing data.

INTERPRETATION AND VISUALIZATION METHODS FOR TRADITIONAL MACHINE LEARNING.

The most general classes of MIV methods are filter and wrapper methods (Kohavi and John 1997). Filter methods consider only the data themselves, whereas wrapper methods incorporate (wrap around) the model. We focus on wrapper methods, as the goal is to understand what the ML models have learned from the data. This section describes interpretation methods designed for traditional ML, though all but impurity importance can be generalized to DL.

Impurity importance. Impurity importance (Louppe et al. 2013; Breiman 2001) is one of the most popular importance methods because it can be computed during training and is implemented in scikit-learn. However, this method works only for tree-based models. It is animated in supplemental Fig. ES3 and defined as follows:

$$I(p) = (1/T)\sum_{t=1}^{T}\sum_{s\in S_{t,p}}(N_s/N)\Delta i(s),$$

where T is the number of trees in the ensemble, $S_{t,p}$ is the set of all splits in tree t that involve predictor p, $\Delta i(s)$ is the decrease in the impurity score (e.g., Gini or entropy) achieved by split s on the training data, N is the total number of training examples, N_s is the number of training examples passed to split s, and $I(p)$ is the resulting importance for predictor p. The most important predictors are those that affect the most examples (higher in tree) and that split the data more effectively (decrease impurity more).

Permutation importance. The permutation method is another approach for ranking predictor importance. Although initially proposed by Breiman (2001) for RFs, it can be used for any traditional or DL model. There are two variations of permutation importance that we denote as single pass (Breiman 2001) and multipass (Lakshmanan et al. 2015). In both variations, the goal is to determine how much performance deteriorates when the statistical link between a predictor x_j and the target variable is broken. This is achieved by randomly permuting x_j over all examples, then comparing the performance of the trained model on unpermuted data to performance on the permuted data. If performance deteriorates significantly when x_j is permuted, this indicates that x_j is important. If performance does not deteriorate significantly, either 1) x_j is generally unimportant or 2) information in x_j is redundant with information contained in the other predictors. For example, if there are two predictors x_j and x_k, such that $x_k = 2x_j$, they are fully linearly dependent (Pearson correlation = 1.0), so either predictor on its own could be deemed unimportant. Multipass importance aims to address this issue. The single-pass and multipass algorithms are precisely stated in supplemental Fig. ES4. Supplemental Figs. ES5 and ES6 describe the algorithms with animations. Both versions are implemented with publicly available code in Jergensen (2019).

Sequential (forward and backward) selection. Sequential selection is another method for ranking predictor importance, but unlike impurity and permutation importance, it can be used to explicitly add or remove predictors from the model. The algorithms for sequential forward and backward selection (SFS and SBS) are shown in supplemental Fig. ES7 and animated in supplemental Figs. ES8 and ES9. The two algorithms are very similar: SFS begins with a climatological model (one that always predicts the mean target value in the training data), containing zero predictors, and adds one predictor at a time. Conversely, SBS begins with a model trained on all predictors and removes one at a time.

SFS may be likened to the song "99 Bottles of Pop on the Wall," where each pop bottle is a predictor. At the kth iteration of SFS, k–1 predictors (bottles) have been selected (taken off the wall and put on the table). The goal is to find the one remaining predictor that, when added to those already selected, yields the best k-predictor model. The stopping criterion in the algorithm (whether cost function J decreases *significantly*) is purposely vague, as there are many ways to implement this. Basically, J should decrease enough to

justify adding another predictor to the model, which increases model complexity.

SBS can be thought of as SFS in reverse. At the kth iteration, k–1 predictors (bottles) have been removed from the model (put back on the wall). The goal is to find the worst remaining predictor (i.e., the one whose removal from the model yields the smallest increase in J). Again, the stopping criterion is purposely vague. Basically, J should increase enough to justify no longer removing the worst predictor from the model.

Generalized versions of SFS and SBS have been proposed (e.g., Stracuzzi and Utgoff 2004; chapter 9 of Webb 2003), which allow both forward and backward steps in the same algorithm or removing several predictors at each step. Also, genetic algorithms can be used to iteratively improve (evolve) the set of selected predictors (e.g., Siedlecki and Sklansky 1993; Leardi 1996). Last, a method called "sufficient input subsets" (Carter et al. 2018) can be used to find the most relevant predictors for one or a subset of examples, without retraining the model.

Partial-dependence plots. While the above methods can reveal the most important predictors for a given problem, they do not indicate how or why each predictor is important. One way to address this would be to visualize the average prediction for each possible value of x_j, the predictor of interest. However, this does not account for nonlinear interactions between x_j and the other predictors. Partial-dependence plots (PDP; Friedman 2001) address this problem by fixing the value of one or more predictors X^* for all examples, passing these new data through the trained model, and averaging the resulting predictions. This averages out the effects of the other predictors. To make the full PDP, the entire process is repeated for a range of values for X^*. Regions of the PDP with nonzero slope indicate where the ML model is sensitive to X^*. Note that this method provides one plot for each predictor, which can be overwhelming with hundreds of predictors.

Related MIV methods attempt to explain the model's prediction for an individual example. Individual conditional expectation (ICE; Goldstein et al. 2015) is the PDP for a specific example. The ICE plot (which can be shown on the same axes as the PDP) identifies clusters of model behavior, which are regions of the predictor space where the model treats examples similarly. Another method is locally interpretable model-agnostic explanation (LIME; Ribeiro et al. 2016), which fits a simple model, such as linear regression, to a set of slightly perturbed examples. The perturbed examples are similar to the example of interest but with slightly altered predictor values. Predictor weights in the simple model are used to explain the prediction.

INTERPRETATION METHODS FOR DEEP LEARNING. All methods presented in the previous section, with the exception of impurity importance, can be adapted for DL. In this work, we demonstrate how the permutation method can be adapted for DL. Although SFS and SBS can also be adapted, they require significant additional computation for the retraining of each model, making them infeasible for most DL applications. For permutation, images from one channel are permuted over all examples. Thus, each example consists of a set of spatially intact maps, but the maps are temporally shuffled (e.g., the temperature field from 1 January 1970 may be matched with the humidity field from 5 April 2012). All interpretation methods presented in this section are implemented in Lagerquist and Gagne (2019); saliency maps are also implemented in Lagerquist (2018). We discuss primarily CNNs in this section, because this is the DL model used in our results. However, the methods presented herein can be applied to other DL models, such as convolutional long-short-term memory (LSTM) and recurrent neural nets.

Saliency maps. Saliency maps (Simonyan et al. 2014) quantify the influence of each input value (i.e., each predictor at each grid point) on the activation of some part of the CNN. This could be the activation of a particular neuron, a group of neurons, or the final prediction from the CNN. Most often it is with respect to an output neuron, whose activation is a predicted probability, p. The saliency of predictor x at grid point (i,j,k), with respect to prediction p, is $\delta p/\delta x(i,j,k)$. Thus, positive (negative) saliency means that the prediction increases (decreases) as $x(i,j,k)$ changes.

One advantage of saliency maps is that they share the dimensions of the input data, which allows them to be viewed as images (the way meteorologists generally prefer to query data) and overlaid with the input data. One disadvantage is that saliency does not necessarily imply importance: salient values are those with which the prediction changes most dramatically, but they are not necessarily most important for the original prediction (Samek et al. 2017). This disadvantage is alleviated by methods such as layerwise relevance propagation (Samek et al. 2017; Montavon et al. 2018) and class-activation maps ("Gradient-weighted class-activation maps" section). Another disadvantage is that saliency is a linear approximation around the actual value of $x(i,j,k)$, meaning saliency indicates

how the model reacts when x is perturbed slightly from the actual value, but not when it is perturbed drastically.

Gradient-weighted class-activation maps. Class-activation maps (CAM; Zhou et al. 2016) quantify the influence of each grid point, rather than each predictor at each grid point, on the predicted probability of a given class p_k. However, CAM works only on a specific type of CNN architecture, so we use a generalization, gradient-weighted class-activation maps (Grad-CAM; Selvaraju et al. 2017). Grad-CAM quantifies the influence of each grid point on p_k, filtered through a given convolutional layer in the network. In other words, at a given depth in the network, Grad-CAM indicates which spatial locations support the prediction of the kth class. For deeper convolutional layers, the class-activation map tends to be smoother (with less small-scale variation) and more localized (with nonzero values in a smaller part of the physical space), reflecting the tendency for deeper layers to learn higher-level abstractions. The ability to leverage representations at different layers is an advantage of Grad-CAM over saliency maps.

Backward optimization. Backward optimization [BWO; or "feature optimization" in Olah et al. (2017)] creates a synthetic input example that maximizes the activation of particular neuron(s), using SGD ("Machine learning" section). Whereas SGD is used during training to update the network weights in a way that minimizes the loss function, it is used during BWO to update input values in a way that maximizes the activation of the given neuron(s). For example, if the task is tornado prediction and we choose to maximize the activation of the output neuron, BWO will create an "optimal tornadic storm." Conversely, if we choose to minimize the activation, BWO will create an "optimal nontornadic storm." Supplemental Fig. ES10 shows an animation of backward optimization, where the goal is to decrease tornado probability for a tornadic storm that initially had very high forecast probability.

Because SGD only adjusts the values in an array, rather than creating the array from scratch, it requires a starting point or "initial seed." Some options are all zeros, Gaussian noise, or a real-dataset example. The advantage of all zeros and Gaussian noise is that the initial seed almost never resembles a real example, so the synthetic example created by BWO is more novel with respect to the initial seed. The advantage of real-data initialization is that the output of BWO is usually more physically realistic. Another way to make the output more physically realistic is to integrate BWO with a generative model (e.g., Goodfellow et al. 2014), which learns to create novel but representative dataset examples [Montavon et al. (2018), who use the term "activation maximization" instead of BWO]. The sensitivity of BWO to the initial seed can be seen as a disadvantage since it does not yield one "perfect" answer, but it can also be seen as an advantage, since BWO can be run with many initial seeds to obtain different answers.

Novelty detection. Novelty detection (Wagstaff and Lee 2018) finds the most novel, or unexpected, image $\mathbf{X}^*$ in a set of images (trial set) with respect to all images in another set (baseline set), then quantifies the novelty of each value in $\mathbf{X}^*$ (i.e., each predictor at each grid point). The algorithm is detailed in supplemental Fig. ES11. As an example of its use, the baseline set could contain nontornadic storms, while the trial set contains tornadic storms. In this case, novelty detection would quantify which tornadic storms are most novel, and which *parts* of these storms are most novel, with respect to the nontornadic ones. The algorithm involves "features," which are inputs to the first dense layer of the CNN (Fig. 1). It is crucial to remember that the CNN extracts only features that aid in the prediction task, so the results of novelty detection are always with respect to the prediction task.

Novelty detection works by using singular-value decomposition (SVD; section 9.3.5 of Wilks 2006) to create a lower-dimensional representation of the image data (feature vector) and an up-convolutional network (upconvnet; Dosovitskiy and Brox 2016) to transform the feature vector back to image space. Loosely, an upconvnet is a backward CNN. The upconvnet allows novelty to be viewed in image space, which is easier for humans to interpret than feature space. The main outputs of novelty detection are the reconstructed image from the CNN's feature space, the reconstructed image from an SVD approximation the same feature space, and the difference between the two.

METEOROLOGICAL DOMAINS. We briefly summarize each domain (prediction task) for which the MIV methods are used. Full descriptions are found in Lagerquist et al. (2018, 2019a), Gagne et al. (2019), McGovern et al. (2018), and Jergensen et al. (2019). We chose deliberately to present results on a wide variety of meteorological domains to show the wide applicability of the MIV methods. This section is kept brief, as specific details are not needed to understand the results fully. Rather, we wanted to highlight the broad applicability of the results

across spatial and temporal scales and different prediction tasks.

Storm-mode classification. We use traditional ML methods (RF, GBRT, and SVMs) to classify storms into three categories: supercell, part of a quasi-linear convective system (QLCS), and discrete storms. The categories and the human-labeled data both come from Thompson et al. (2012) and Smith et al. (2012). We use data from the years 2003–11. Predictors for this task include radar statistics derived from the Multi-Year Reanalysis of Remotely Sensed Storms (MYRORSS; Ortega et al. 2012) and environmental data from the Rapid Update Cycle (RUC; Benjamin et al. 2004). Models are trained with ninefold cross validation, where each fold is 1 year.

Precipitation type. We apply the same traditional ML methods for predicting winter precipitation type. The four precipitation types are rain, freezing rain, snow, and ice pellets. Labels are Meteorological Phenomena Identification Near the Ground (mPING; Elmore et al. 2014) reports from October 2014 to March 2015. The predictors are statistics derived from RUC proximity soundings. Specifically, soundings are taken from the nearest grid point to the mPING report at 6-, 12-, and 18-h lead times. Although soundings from the three lead times are not independent, they increase the size and variability of the dataset, which is crucial given that the time period is only one winter. The three forecasts have the same valid time, so differences among them are due solely to differences among the three model runs. The results in this paper come from training on the classic warm-nose sounding, characterized by an elevated warm (melting) layer above a cold (freezing) layer at the surface. This type of sounding contains two freezing layers, one above the elevated warm layer and one below, and is the type most commonly associated with freezing rain and ice pellets.

Tornado prediction. We use a CNN to forecast tornadogenesis. Specifically, for each storm object (one thunderstorm cell at one time), the CNN is applied to a storm-centered radar image and proximity sounding, with the goal of predicting whether or not the storm will generate a tornado in the next hour. Radar images come from the GridRad dataset (Homeyer and Bowman 2017), a mosaic of all Next Generation Weather Radar (NEXRAD) scans in the CONUS. The GridRad data used here have a horizontal resolution of 0.02°, vertical resolution of 0.5 km up to 7 km above sea level and 1.0 km aloft, and temporal resolution of 5 min. Storm-centered 2D grids (e.g., Fig. 9) are 32 × 32, interpolated to 1.5-km horizontal resolution, and rotated so that storm motion is toward the right. The composites contain 12 variables: minimum, mean, and maximum reflectivity from 1 to 3 km above ground level; minimum, mean, and maximum 1–3-km radial velocity spectrum width; minimum, mean, and maximum 2–4-km vorticity; and minimum, mean, and maximum 5–8-km vorticity. The choice of these variables is based on previous work by Sandmael and Homeyer (2018), showing that they discriminate well between tornadic and severe nontornadic storms.

Soundings come from the RUC model before 1 May 2012 and the Rapid Refresh (RAP; Benjamin et al. 2016) otherwise. In general, interpretation results are shown only for radar data, as results for soundings have been noisy. Tornado reports from *Storm Data* (National Weather Service 2016) are used to determine when/if a storm undergoes tornadogenesis.

Hail prediction. We use CNNs to predict large hail in simulated thunderstorms (Gagne et al. 2019) from the National Center for Atmospheric Research (NCAR) convection-allowing ensemble (CAE; Schwartz et al. 2015). The target variable is based on the storm's maximum future hail size ("yes" if ≥25 mm in diameter, "no" otherwise), according to the Thompson microphysics scheme (Thompson et al. 2004, 2008). This is a perfect-model experiment, because the target variable comes from a simulation rather than true observations. The goal is to identify storm-scale and environmental features that promote *simulated* hail growth.

The predictors are storm-centered grids of five variables (temperature, dewpoint, u wind, v wind, and geopotential height) at three pressure levels (850, 700, and 500 hPa). The grids are 32 × 32 and share the 3-km grid spacing of the NCAR CAE. Each CNN trained for this problem has three strided-convolution layers (which combine the convolution and pooling operations into one layer), with either ReLU or leaky-ReLU activation (Maas et al. 2013), followed by a dense layer with sigmoid activation (cf. Fig. 1).

RESULTS. We organize results by MIV method, beginning with traditional ML and moving to DL.

Ranking and selecting important predictors. Figure 2 compares importance rankings for tree-based models, using both impurity and permutation importance for the tasks of storm-mode classification and winter

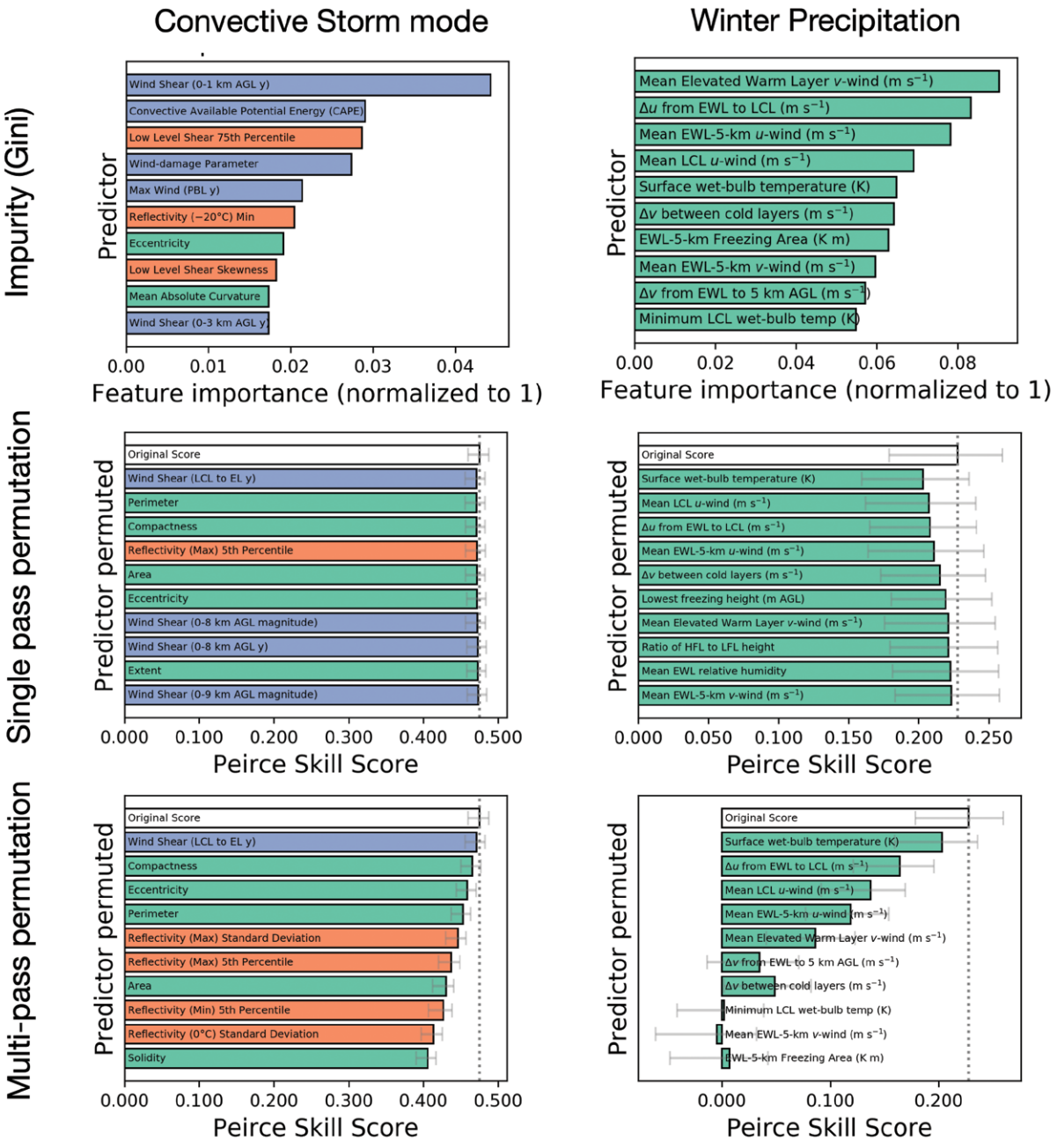

Fig. 2. Comparison of importance rankings for tree-based models. The prediction tasks are (left) storm-mode classification and (right) winter precipitation type. Each panel shows the 10 most important predictors, with importance decreasing from top to bottom. Radar variables are in orange, sounding parameters are in purple, and other predictors (on left, shape parameters describing the storm outline) are in green. Large colored bars show the mean, while error bars show the 2.5th and 97.5th percentiles, for 1,000 bootstrap replicates of the validation set.

precipitation. It is important to understand both the meteorological significance of the predictors and the difference in importance rankings among the methods. This is especially important for researchers who may have previously used only one importance ranking.

For winter precipitation type, the two versions of permutation agree on the top four predictors (with a difference in ordering for the second and third): surface wet-bulb temperature, mean u wind in the lowest cold layer (LCL), u-wind difference between the LCL and elevated warm layer (EWL), and mean

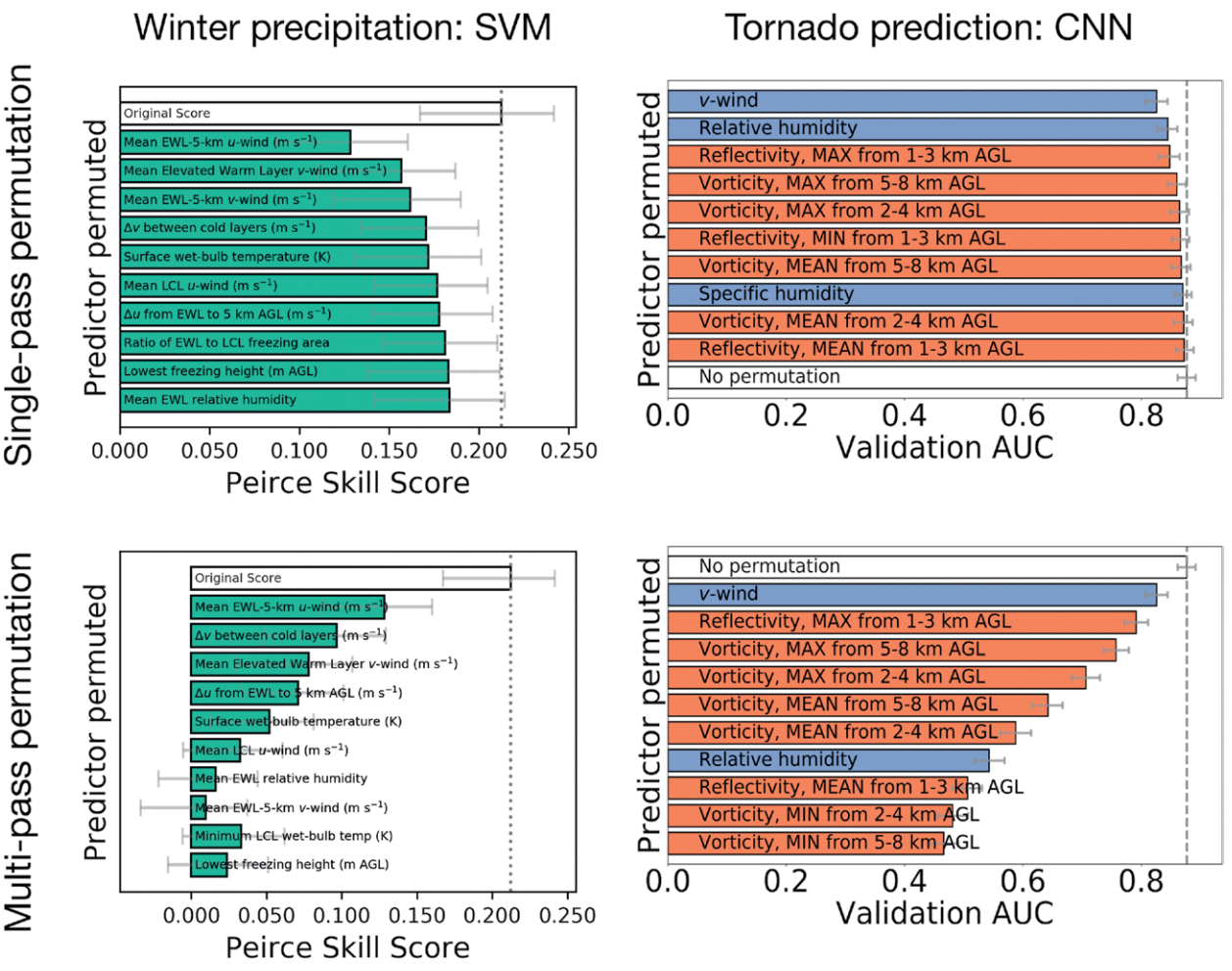

FIG. 3. Comparison of permutation methods (single and multipass) for non-tree-based models. (left) Radial-basis-function SVM for winter precipitation type. (right) CNN for tornado prediction. Each panel shows only the 10 most important predictors, with importance decreasing from top to bottom. All other details (color scheme and error bars) are as in Fig. 2.

u wind from the EWL to 5 km above ground level (AGL). The importance of these predictors makes sense meteorologically. For example, (isobaric) wet-bulb temperature is the temperature that an air parcel would be if it were cooled by evaporating water into it at constant pressure. If the surface temperature is >273.15 K, freezing rain is impossible and rain becomes much more likely. There are more substantial differences between impurity importance and permutation importance. For example, the top predictor (surface wet-bulb temperature) using permutation is fifth-most important according to impurity, while the most important predictor for impurity (mean *v* wind in the EWL) is fifth- or seventh-most important according to permutation.

For storm-mode classification, the two versions of permutation importance agree on three of the top four predictors (with a slight difference in ordering): the *y* component of lifting condensation level-to-equilibrium level (LCL–EL) wind shear, perimeter, and compactness. In general, the most important predictors are reflectivity statistics (spatial statistics based only on grid cells inside the storm), environmental wind shear (in the proximity sounding), and shape parameters. These results are broadly consistent with what we know about storm mode, especially the difference between supercells and other modes: supercells tend to be less elongated (lower eccentricity) than QLCS storms, with higher reflectivity and higher wind shear. Again, the two versions of permutation agree more with each other than with impurity importance. However, impurity importance still emphasizes shape parameters, reflectivity, environmental wind shear, and low-level shear. This last variable is radar-derived azimuthal shear from 0 to 2 km above ground level and is greater in supercells, due to rotation in the mesocyclone.

Permutation can also be used to rank predictors for non-tree-based models, as shown in Fig. 3. For

winter precipitation, the two versions agree on three of the top four (and five of six) predictors for SVMs: mean u wind from the EWL to 5 km AGL, mean v wind in the EWL, and v-wind difference between the two cold layers (below and above the EWL). However, these are mostly disparate from the top predictors in Fig. 2, for the RF. This suggests that the RF and SVM "care about" different predictors. This is expected, since the two models internally are very different. According to the error bars in Figs. 2 and 3, the performance of the two models is statistically similar, either before permutation or after any number of permutations. Thus, there is no reason to give more credence to the permutation-importance results of one ML model over the other. This underscores the importance of using several ranking methods and considering general types of predictors (e.g., sounding and shape statistics) rather than just single predictors. The different rankings would also be easily explainable if there were many linearly dependent predictors. However, the predictors for this task were preprocessed to remove any absolute Pearson correlation > 0.7 in the data.

For tornadogenesis (Fig. 3), the most important predictor [ranked by area under the receiver operator curve (AUC); Metz 1978] is v wind (meridional wind in the proximity sounding). The third- to fifth-most important predictors in the single-pass method (maximum low-level reflectivity, midlevel vorticity, and low-level vorticity) match the second- to fourth-most important predictors, respectively, in the multipass method. Perhaps the most striking difference is that RH (relative humidity in the sounding) is ranked second by the single-pass method but seventh by the multipass method. This suggests that in the multipass method, after the first predictor (v wind) has been

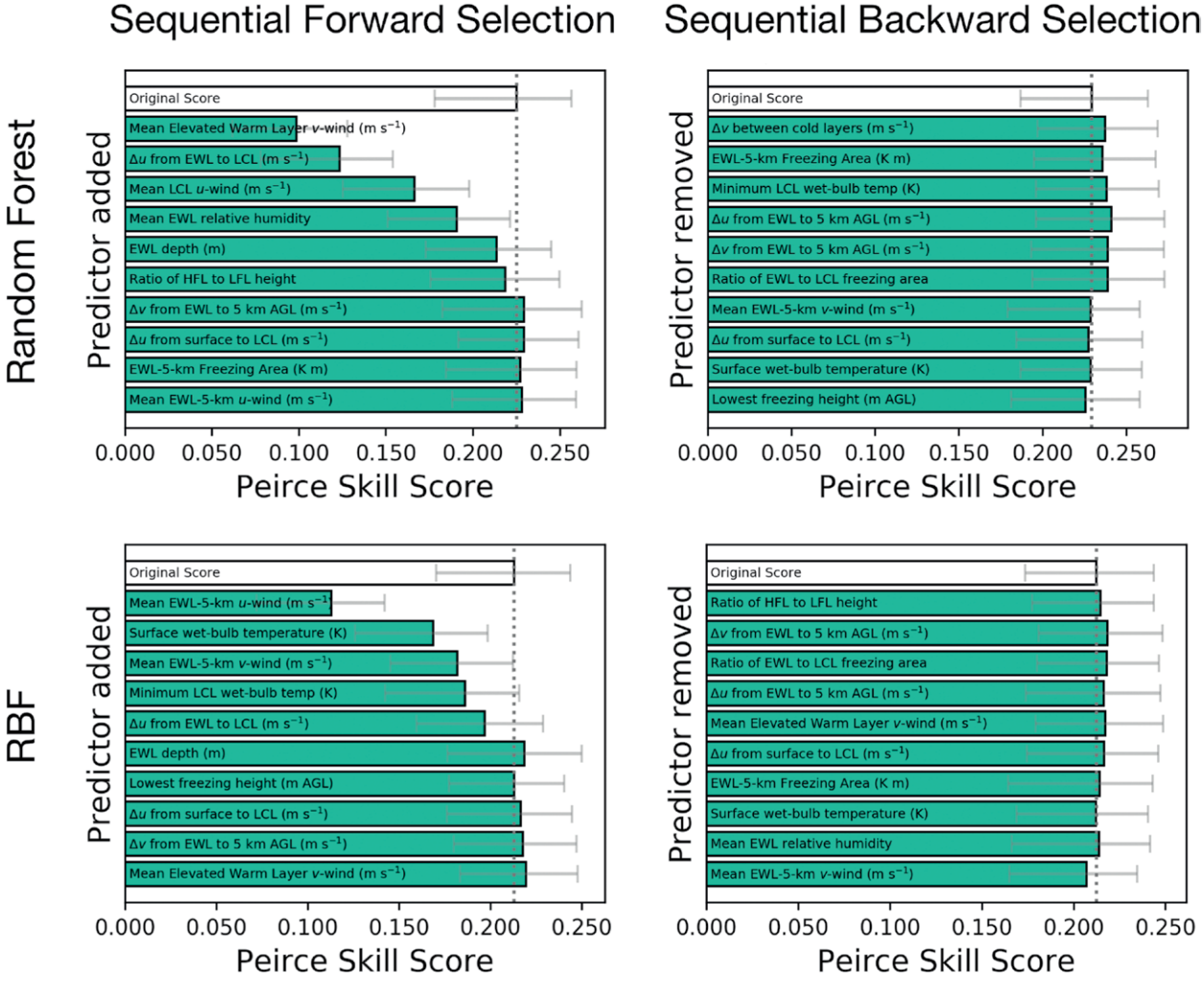

Fig. 4. Sequential-selection results for winter precipitation type for (top) a random forest and (bottom) a radial-basis-function SVM. (left) For SFS, the first predictor selected is at the top and the last selected is at the bottom. (right) For SBS, the first predictor *removed* is at the top and the last removed is at the bottom.

permuted, RH is no longer the most important predictor. This is counterintuitive, as RH is less dependent on v wind (Pearson correlation of 0.11) than maximum low-level reflectivity on v wind (0.43). However, Pearson correlation is linear and based on the entire dataset. It is possible that, for a small partition of the dataset with a strong effect on AUC, there is more dependence between RH and v wind, causing the two variables to be more redundant than indicated by the Pearson correlation.

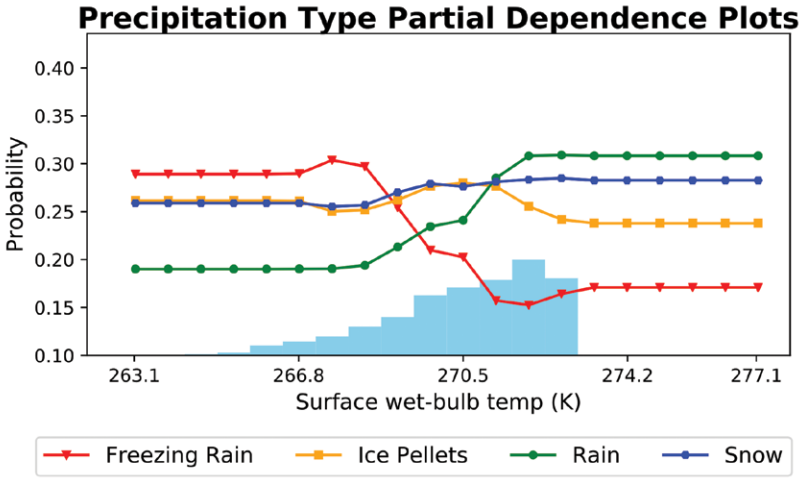

Fig. 5. Partial-dependence plots for the probability of each precipitation type, given two predictors in the random forest. Blue histograms show the distribution of each predictor in the training data.

In the single-pass method, after the top few predictors, there is very little difference in importance among the rest. This is because the other 16 predictors, which contain the vast majority of useful information, are still intact. Conversely, in the multipass method, AUC decreases substantially with each successive predictor permuted until it reaches ~0.5 after the ninth predictor (minimum low-level vorticity), which is the AUC for a completely random model. Overall, both methods suggest that reflectivity and vorticity are the most important radar predictors, while v wind and relative humidity are the most important sounding predictors.

SFS and SBS are used to explicitly select predictors to keep in an ML model. Figure 4 shows results for winter precipitation type, based on both the RF and SVM models. Results for storm mode are not shown, because SBS would take hundreds of hours with the hundreds of predictors used for this task. For the RF, while forward and backward selection do not agree precisely, their ranking of predictors is similar. For the SVM, forward and backward selection agree on only two of the top five (and five of the top eight) predictors. Regularization of the SVM may affect these results, something that requires further investigation. RFs perform predictor selection internally (by choosing the best predictor at each split point; "Machine learning" section), but SVMs do not: if it receives 17 predictors it must fit a 17-dimensional hyperplane, which can lead to unstable results (Vapnik 1995).

Partial-dependence plots. A partial-dependence plot for precipitation type for the most important predictors in the RF is shown in Fig. 5. The easiest curves to interpret are those for rain and freezing rain. As surface wet-bulb temperature increases from approximately 267 to 273 K, rain probability increases sharply while freezing-rain probability decreases sharply. Freezing rain is impossible when surface temperature exceeds 273.15 K, because it requires that rain fall as liquid and freeze upon contact with the surface. The fact that the RF and SVM create freezing rain at T_w well below freezing and freezing rain at T_w well above freezing is a particular characteristic of the NWP model errors.

Saliency maps. Saliency maps for tornado prediction are shown in Fig. 6. For the sake of brevity, these maps include only four of the 12 radar variables listed in the "Tornado prediction" section. Each row is a composite over 100 examples (one example indicates one storm at one time) in the validation period: the best true positives (tornadic examples with the highest forecast probabilities), worst false alarms (nontornadic examples with the highest probabilities), worst misses, and best correct nulls. Composites are created with the method of probability-matched means (PMM; Ebert 2001). We initially tried using simple means, but the resulting composites were unrealistic due to spatial offsets among storms in the composite. We have examined interpretation outputs for individual storms (e.g., supplemental Fig. ES10), and the results are conceptually similar to the composites, so we are not overly concerned about artifacts introduced by PMM. In the future we will add local PMM (Clark 2017).

For the best true positives, the composite radar image looks like a supercell (e.g., Kumjian and Ryzhkov 2008), with a large core of reflectivity > 55 dBZ; a slight hook echo on the right flank (more visible in the top-left

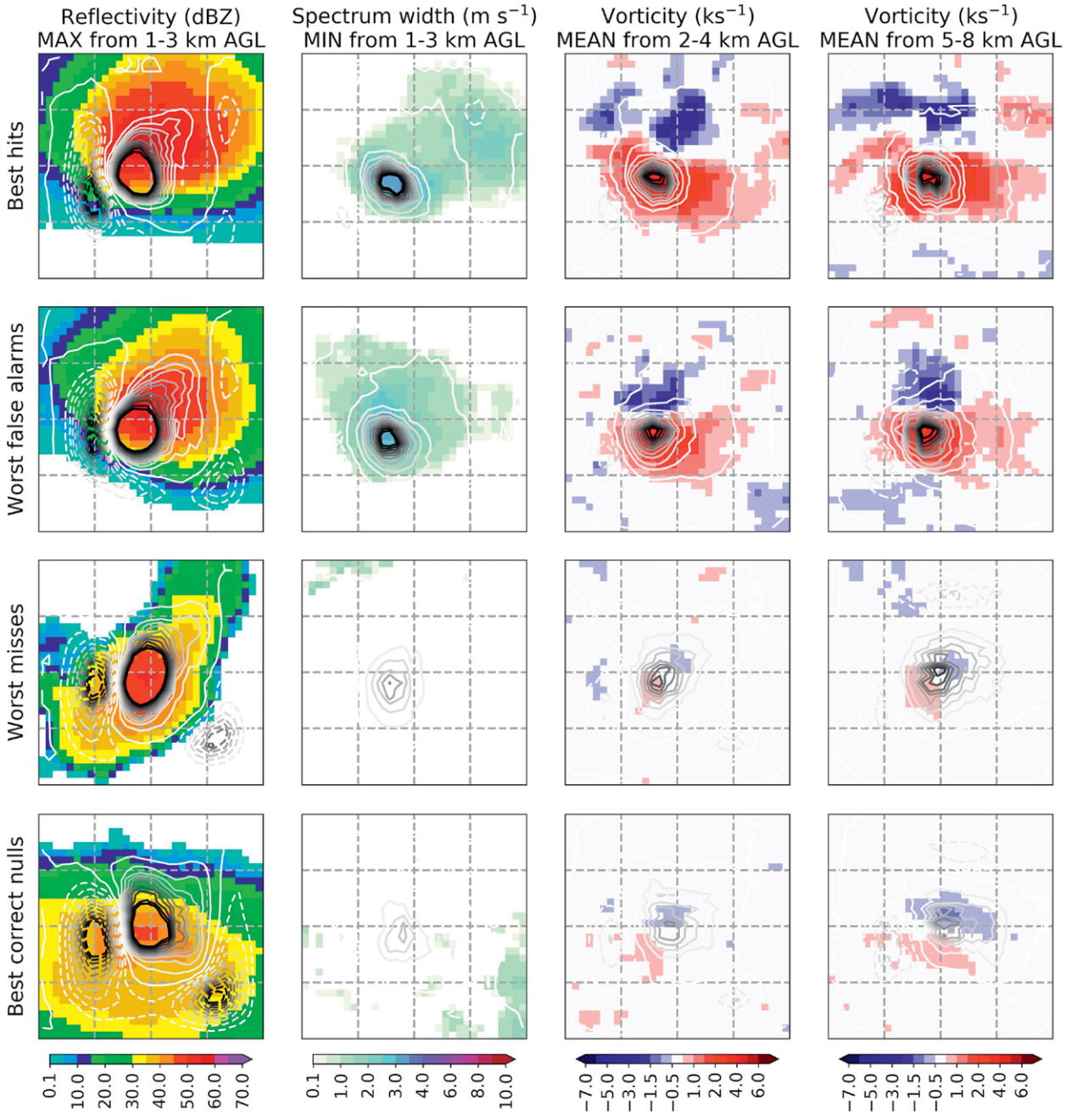

Fig. 6. Composite saliency maps for the 100 best hits, worst false alarms, worst misses, and best correct nulls. Storm motion is to the right. Heat maps represent four of the 12 input fields (predictors): (left to right) maximum reflectivity from 1 to 3 km AGL, minimum velocity-spectrum width from 1 to 3 km AGL, mean vorticity from 2 to 4 km AGL, and mean vorticity from 5 to 8 km AGL. Line contours represent saliency. Positive values, which indicate that tornado probability increases with the underlying predictors, are shown with solid contours, negative values are shown with dashed contours, and darker colors indicate larger absolute values.

panel of Fig. 9); and large maxima of low-level reflectivity, low-level spectrum width, low-level vorticity, and midlevel vorticity in the mesocyclone. According to the composite saliency map, tornado probability increases strongly with all four of these maxima and decreases strongly with reflectivity behind the storm, especially near the mesocyclone. This latter relationship suggests that tornadoes are more likely when the storm is more isolated from surrounding deep convection and the rear-flank downdraft is not too cold (due to evaporative cooling; e.g., Markowski et al. 2002), concepts familiar to human meteorologists.

Adebayo et al. (2018) discuss three "sanity checks," which ensure that saliency maps reflect meaningful

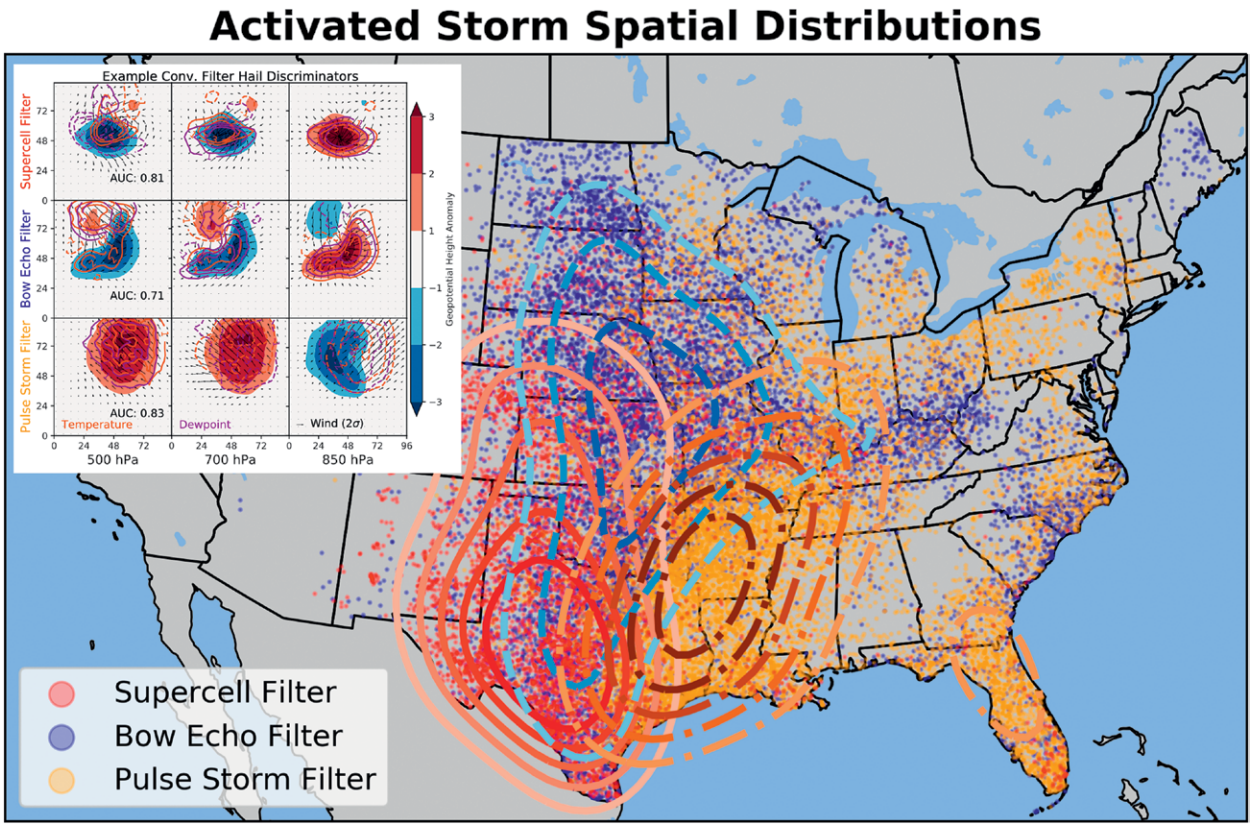

Fig. 7. The inset shows composite saliency maps for selected neurons in the last (deepest) convolutional layer of the hail-prediction CNN. These neurons are preferentially activated by storms with human-identifiable morphologies. Anomalies are in standard-deviation units, based on means and standard deviations over the training data. "AUC" for neuron *n* is the area under the ROC curve, computed on all hailstorms, for how well *n* identifies severe hailstorms when activated. The main panel shows the spatial distribution of storms that strongly activate each filter. Adapted from Gagne et al. (2019).

relationships learned by the model, rather than patterns that exist in all data, such as the "Buell patterns" that often appear in principal-component analysis (Richman 1986). We have implemented the edge-detector test and supplemental Fig. ES12 shows that the "saliency maps" produced by an untrained edge detector are markedly different than those produced by the trained model.

For the worst false alarms, the composite radar image is very similar to the best hits, except that there is no discernible hook echo, a smaller reflectivity core, and smaller maxima of all four variables. The composite saliency map is also similar to the best hits, which indicates that if these maxima were increased to their levels in the best hits, tornado probability would increase strongly. For the worst misses, the composite radar field looks very different than the best hits and worst false alarms. The reflectivity core has a more linear structure, which suggests that many of the 100 storms are part of a QLCS. This makes sense, given that QLCS tornadoes are often missed by human forecasters (Table 2; Brotzge et al. 2013). Minimum low-level spectrum width is near zero throughout most of the domain, possibly because these storms tend to be more elevated (with bases higher aloft than the best hits and worst false alarms). Also, maxima of the two vorticity fields are only 0.5–1.0 ks^{-1}, about 10 times smaller than for the best hits or worst false alarms. Tornado probability increases strongly with reflectivity, slightly with spectrum width, and moderately with vorticity, in the reflectivity core. Tornado probability also decreases strongly with reflectivity behind the core and moderately with reflectivity in front of the core, indicating a preference for isolated convection. For the best correct nulls, the composite radar image and saliency map look roughly similar to the worst misses. The main differences are that the reflectivity core is weaker and the storm is more elongated in the direction of motion.

For the hail-prediction task, composite saliency maps for selected CNN neurons are shown in Fig. 7 (inset). The top neuron is activated by supercell-like

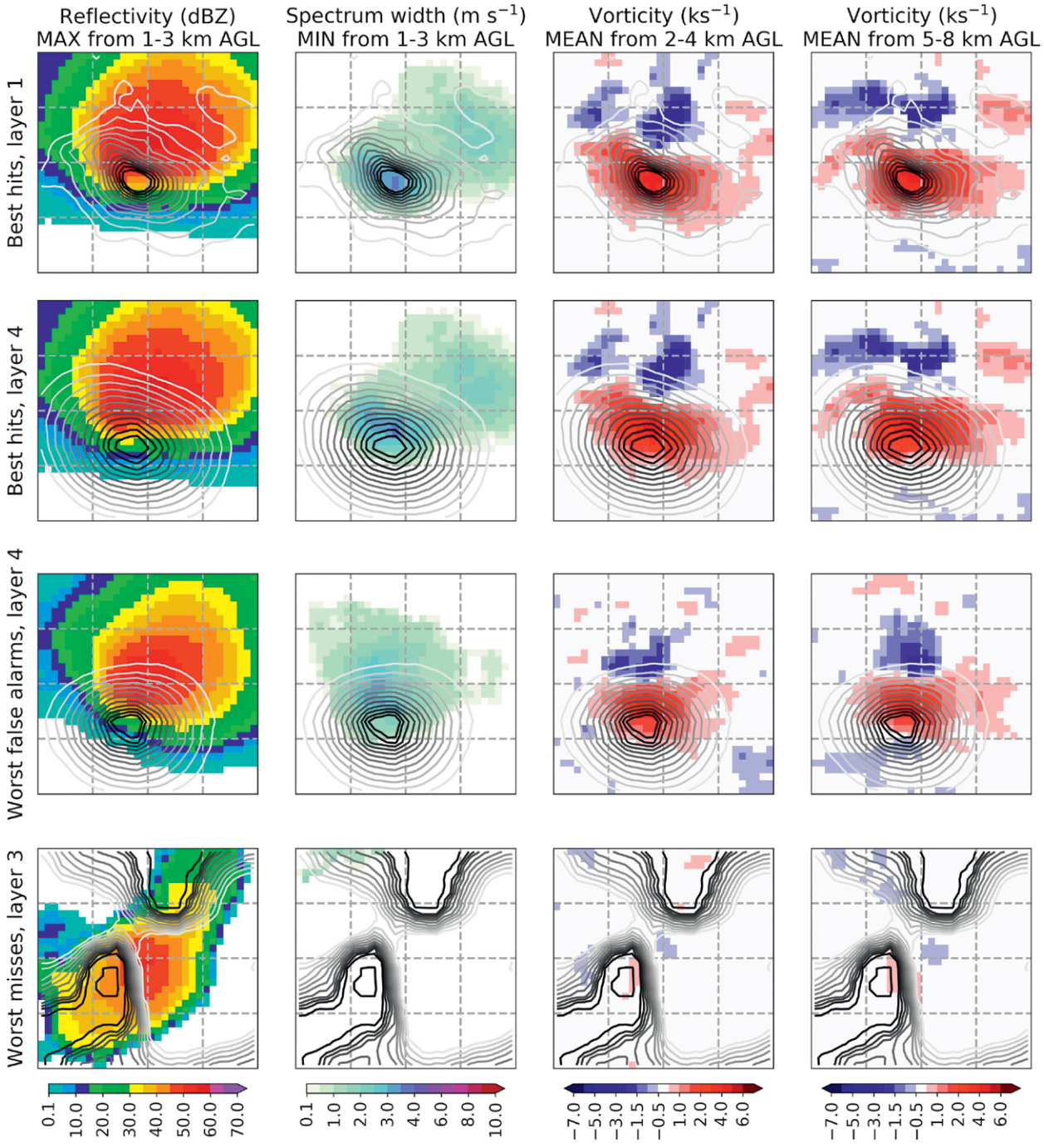

Fig. 8. Composite Grad-CAM (gradient-weighted class-activation maps) for the 100 best hits, 100 worst false alarms, and 100 worst misses, according to different convolution layers. Storm motion is to the right. Layer 1 is the shallowest, and layer 4 is the deepest (Fig. 1). Heat maps represent input fields (predictors), as in Fig. 6, while line contours represent class activation. Darker colors indicate that the underlying spatial location has a greater positive influence on "yes" (tornado) predictions. Negative influences cannot be shown with Grad-CAM. For the 100 worst misses, output is shown for layer 3 rather than layer 4, because class-activation maps produced by layer 4 are all zeros.

storms, with a rounded shape and rotational wind field. The middle neuron is activated by bow-echo-like storms, with an elongated shape, strong low-level convergence, and little rotation. The bottom neuron is activated by pulse-type storms, with neither an exceptionally rounded nor exceptionally elongated shape and an outflow-dominant low-level wind field. The geographic map in Fig. 7 shows the spatial

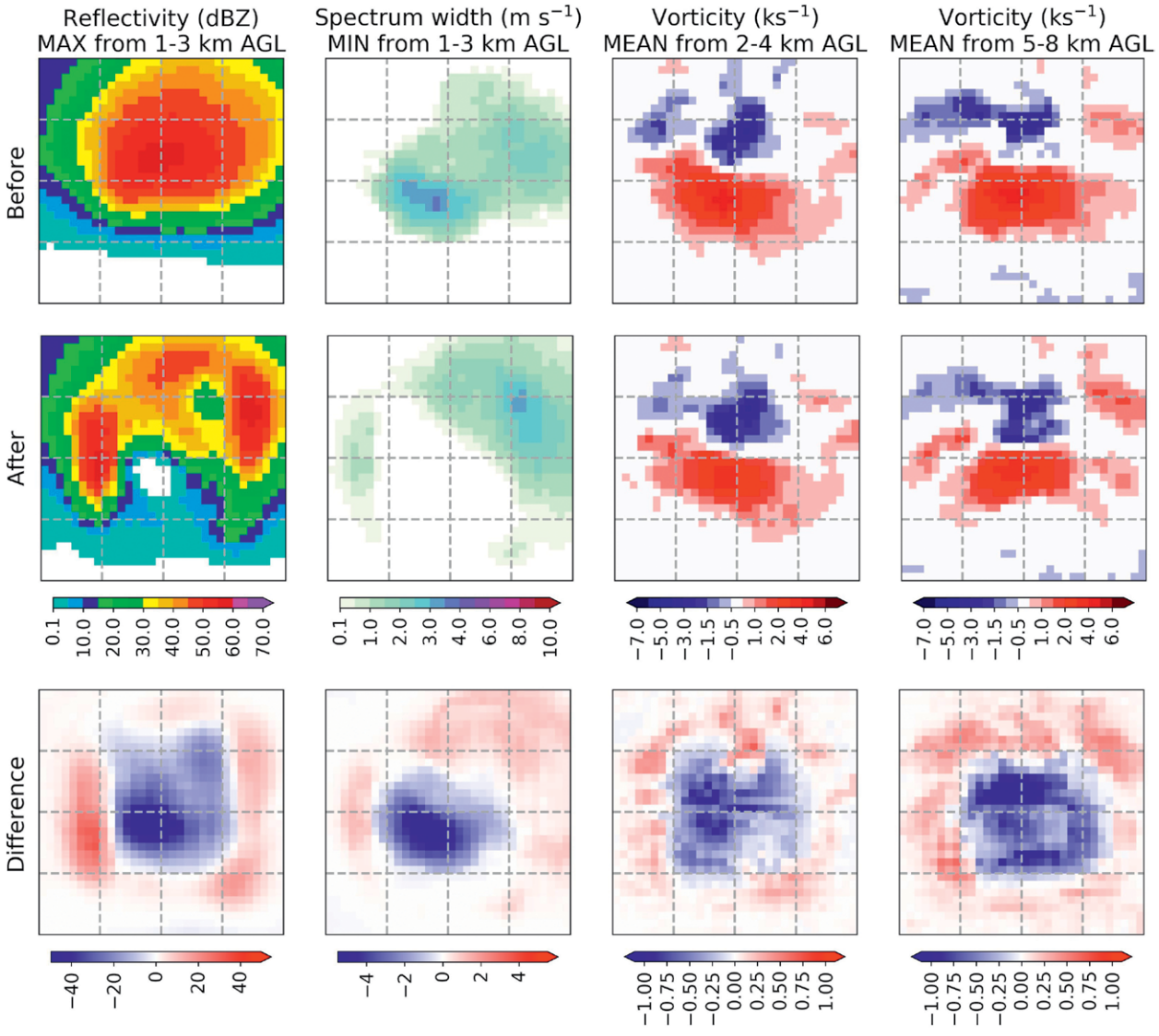

Fig. 9. Composite backward-optimization results for the 100 best hits. Storm motion is to the right. In this case the goal of backward optimization was to *minimize* tornado probability. (top) Input fields (before optimization); (middle) output fields (after optimization), and (bottom) the increments made by backward optimization (after minus before).

distributions of storms that strongly activate the three neurons.

Class-activation maps. CAMs for tornado prediction, produced by Grad-CAM, are shown in Fig. 8. These maps show the most important grid cells for tornado prediction—that is, those that most strongly support a "yes" forecast. The first CAM for the best true positives, produced by the first (shallowest) convolution layer, suggests that the most important locations are in the mesocyclone, collocated with the slight hook echo and maxima in spectrum width and vorticity. Outside of this region, class activation decreases sharply. Contours are elongated along the right flank, indicating that class activation decreases less sharply along the right flank than perpendicular to it. This makes sense, as the right flank is adjacent to the storm's inflow environment, to which tornadogenesis is highly sensitive [e.g., review in first paragraph of Wade et al. (2018)]. The second CAM for the best hits, produced by the fourth (deepest) convolution layer, is similar to the first CAM, except that contours are smoother and nonzero activations are more confined to the mesocyclone. This makes sense, given that (i) inputs to the fourth convolution layer have coarser resolution (3 vs 1.5 km) and (ii) deeper layers learn higher-level abstractions, increasing their ability to selectively focus on a small part of the image (e.g., the

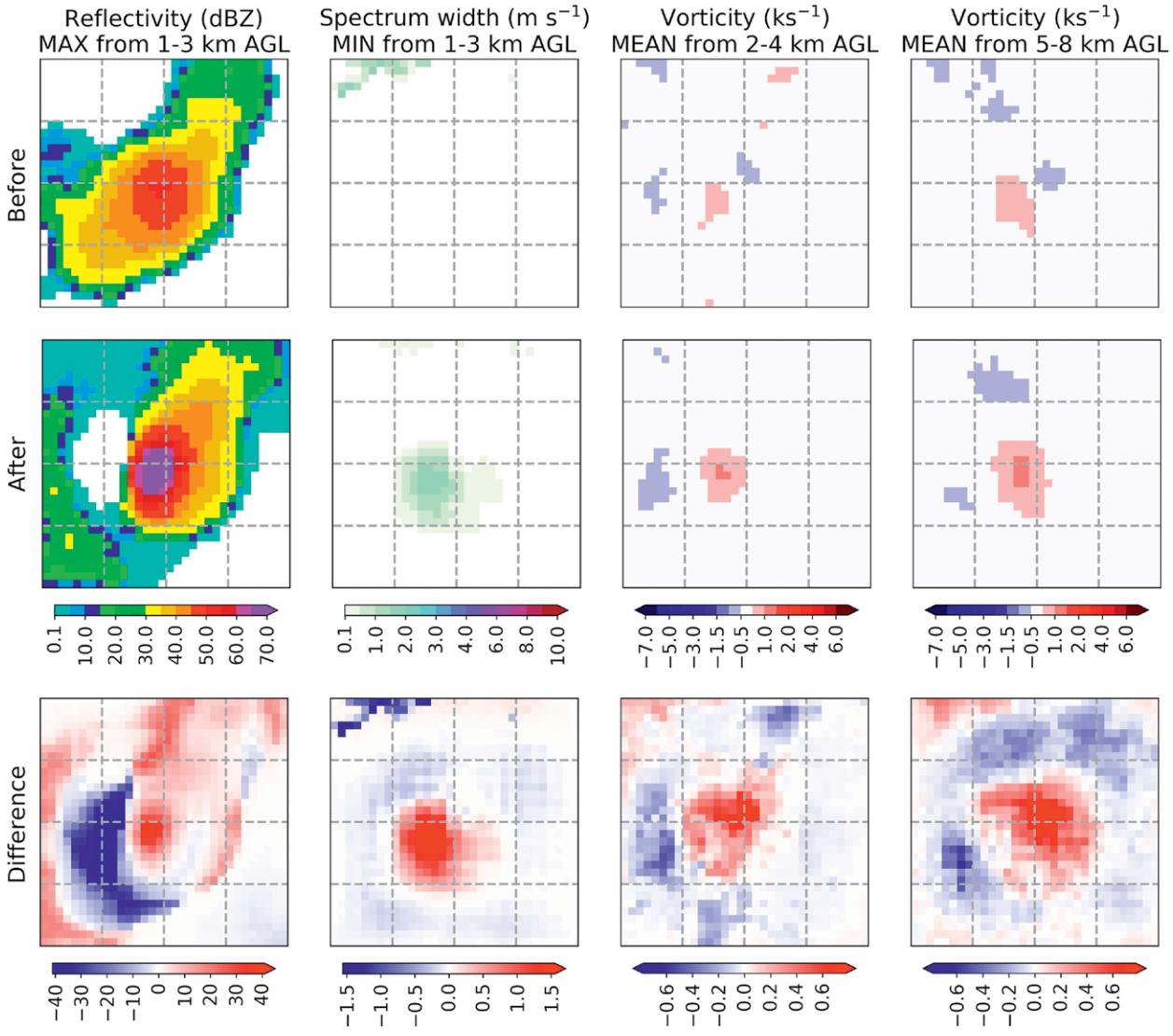

Fig. 10. Composite backward-optimization results for the 100 worst misses. Storm motion is to the right. In this case the goal of backward optimization was to *maximize* tornado probability (otherwise as in Fig. 9).

main vorticity maximum in the reflectivity core and not the secondary one in the forward flank). There is a slight offset between class-activation maxima for the first and fourth convolution layers, which is probably due to upsampling from dimensions of 10×10 to 32×32.

The CAM for the worst false alarms is similar to the best hits, with two main exceptions. First, nonzero activations cover a smaller area, consistent with the reflectivity core covering a smaller area. Second, contours are elongated in the direction of storm motion, rather than along the right flank. This indicates that the right flank is less supportive of "yes" forecasts in the worst false alarms than in the best hits, which makes sense, as vorticity maxima for the best hits extend farther along the right flank. Finally, the CAM for the worst misses is produced for the third (second deepest) convolution layer (Fig. 1). We show the third convolution layer, instead of the fourth, because CAMs for the fourth layer are all zeros. This makes sense, given that forecast probabilities for the worst misses (by definition) are very low. According to the CNN, the radar image as a whole does not support a "yes" forecast. Progressing deeper in the network, from the first to fourth convolution layer (not shown), the area of nonzero class activations is pushed away from the center until all activations become zero.

Backward optimization. BWO results for tornado prediction are shown in Figs. 9 and 10. In Fig. 9, BWO is used to adjust each of the 100 best true positives (as the seed) with the goal of decreasing tornado probability. BWO decreases the CNN's forecast probability from near one to near zero. Conversely, in Fig. 10,

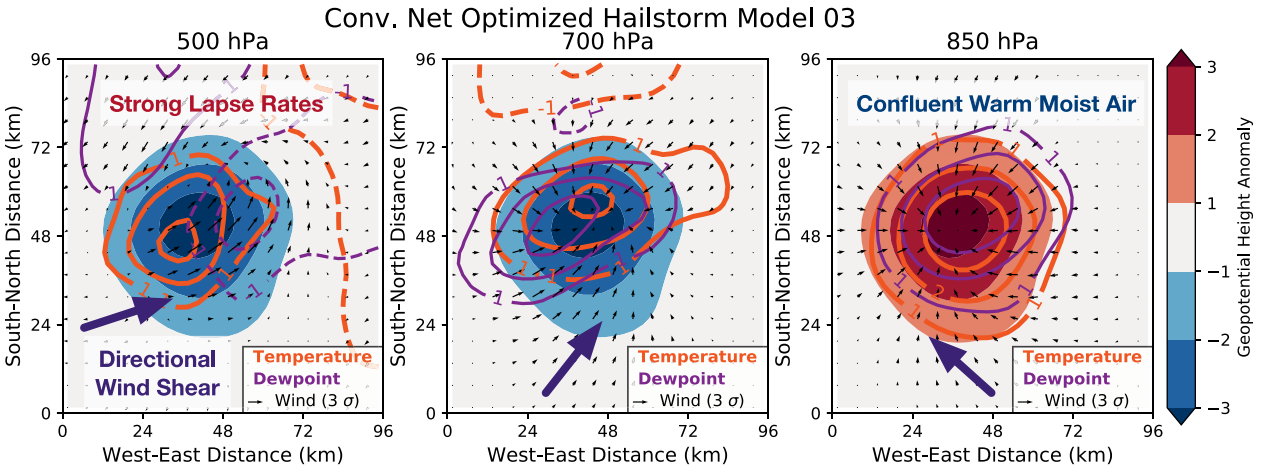

Fig. 11. Backward-optimization results for hail prediction. An all-zero array was optimized to increase large-hail probability, resulting in the synthetic storm (optimal hailstorm) shown. Anomalies are in standard-deviation units, based on means and standard deviations over the training data. Adapted from Gagne et al. (2019).

BWO is used to adjust the worst misses (as the seed), with the goal of increasing tornado probability. Here, BWO increases the probability from near zero to near one. In general, BWO has the greatest effect on low-level reflectivity and spectrum width (making changes up to 40 dBZ and 4 m s^{-1}, respectively), with a much subtler effect on the vorticity fields (making changes of ~10^{-3} s^{-1}). However, as shown in the reflectivity fields, BWO does not necessarily produce realistic output. It is possible that more realistic output could be encouraged by adding physical constraints to the loss function, as is sometimes done in objective analysis [e.g., the geostrophic constraint used in Panofsky (1949), Bergthórsson and Döös (1955), and Cressman (1959)].

BWO for hail prediction is shown in Fig. 11. The initial seed (an array of all zeros) has been adjusted by the CNN to maximize large-hail probability. The output (synthetic storm) includes a positive height anomaly at 850 hPa, with negative height anomalies at 700 and 500 hPa. This height-anomaly gradient is associated with a high lapse rate (strong increase of temperature with pressure), which can lead to more instability and a stronger updraft. The synthetic storm also includes positive temperature and dewpoint anomalies at 850 and 700 hPa, along with confluent winds, indicating that warm and moist air is flowing into the storm. Finally, winds in the inflow region (bottom of the map) rotate clockwise with height, which is favorable for right-moving supercells (Bunkers et al. 2000).

Novelty detection. Novelty detection for tornado prediction is shown in Fig. 12. Each row is a PMM composite over the 100 most novel examples for which the storm undergoes tornadogenesis in the next hour in the validation period. Both the actual feature vector and its SVD reconstruction are projected to image space by the upconvnet, which allows the input (radar image) and output (novelty map) to be viewed in the same space. The most novel or unexpected parts of the examples shown are low reflectivity to the storm's right, which indicates a lack of deep convection in the inflow environment; high spectrum width in the mesocyclone and reflectivity core; high low- and midlevel vorticity to the storm's right; and low midlevel vorticity on the left side of the reflectivity core.

Although the upconvnet cannot exactly map storms from feature space to image space, it can highlight novel or interesting areas of the input for further examination. One must be careful to ensure that artifacts of the upconvnet, such as the positive low-level vorticity anomalies to the storm's right and in the forward flank, are recognized as such. The novel regions can be used for knowledge discovery and further hypothesis testing.

DISCUSSION AND FUTURE WORK.

This paper synthesizes and analyzes ML MIV methods and demonstrates their use for various meteorological domains. Table 1 provides a high-level summary, listing the advantages and disadvantages of each method and Table 2 summarizes when a user should choose each method. As ML continues to gain popularity in meteorology and other physical sciences, it is crucial for practitioners to understand the trade-offs inherent in the models themselves and the MIV methods used to explain them. It is also important to understand the computational trade-offs of these methods. Some

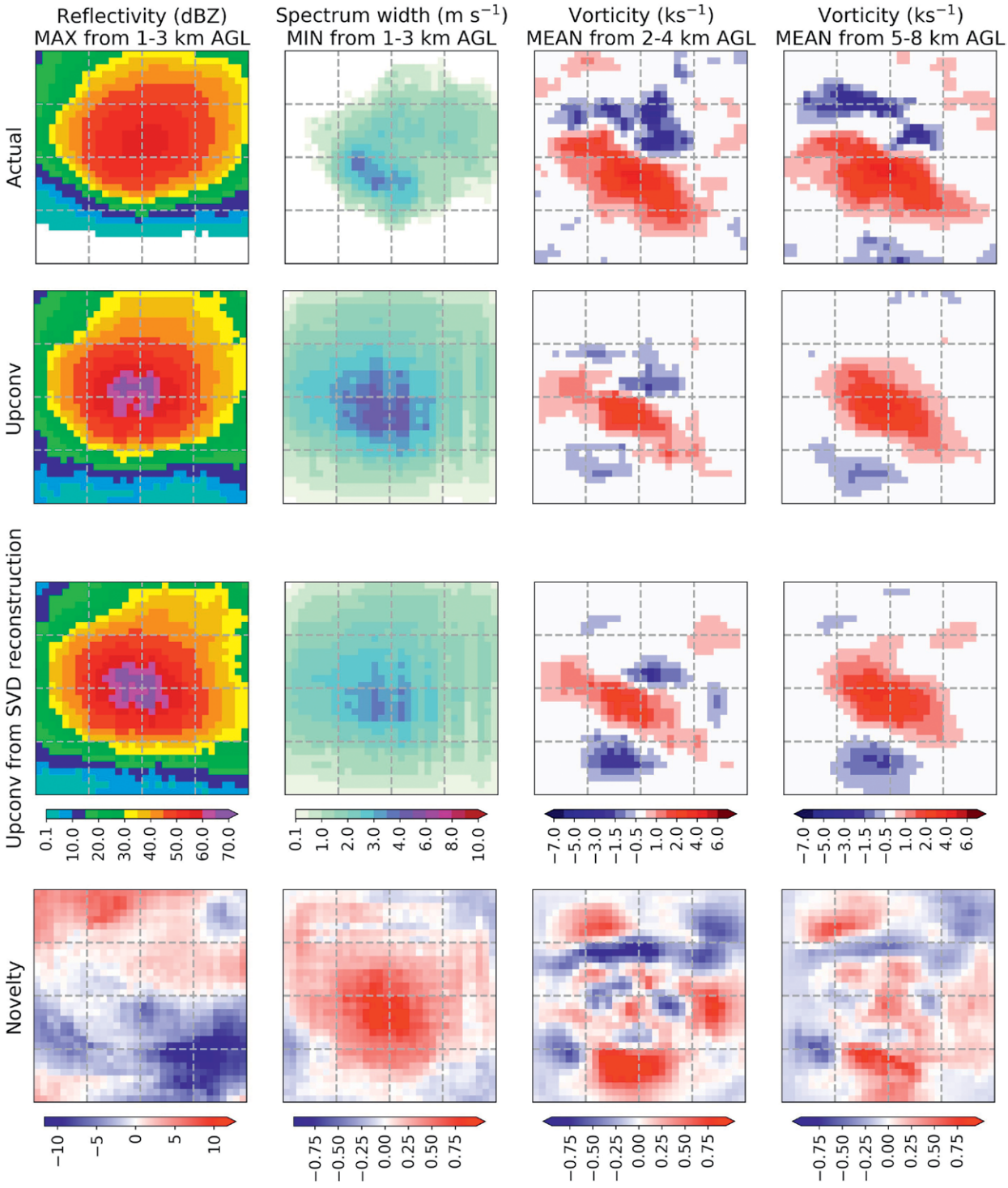

Fig. 12. Novelty detection for the 100 most novel tornadic examples in the validation period. Storm motion is to the right. (top to bottom) Actual storms, upconvnet projection of the storms' feature vectors back to image space, analogous, except for SVD reconstructions of the feature vectors, and the novelty map (first upconvnet projection minus second).

methods are efficient, while some may take additional supercomputing time, meaning that users need to decide if additional computational effort is worth the potential insights gained.

For example, if the user wants to identify the most important predictors for the ML model, the most computationally efficient approach is impurity importance. However, impurity importance has a

Table 1. Advantages and disadvantages of MIV methods.

Method	Advantages	Disadvantages
Impurity importance ("Impurity importance" section)	Succinct list of predictors Computed at training time	Does not explicitly select predictors (underlying tree does selection) Does not explain why something is important Works only for tree-based methods
Permutation importance ("Permutation importance" section)	Succinct Model agnostic (can be applied to any ML model) Single-pass method can be easily parallelized	Greedy algorithm that chooses one predictor at each step, which limits the search of solution space and also make it difficult to compare across runs if the underlying ML model is brittle Multipass is computationally expensive Does not explain why something is important
Sequential selection ["Sequential (forward and backward) selection" section]	Succinct Model agnostic Selects most relevant predictors Generalized versions allow algorithm to be somewhat nongreedy	Computationally inefficient (model is retrained many times) Usually implemented as a greedy algorithm Does not explain why something is important
Partial-dependence plots ("Partial-dependence plots" section in "Interpretation and visualization methods for traditional machine learning" section)	Model agnostic Explains how predictor *x* is important: i.e., how output changes with *x* over range of *x*	Difficult to extend to deep learning Inefficient for multivariate interactions Potentially overwhelming for the human when used for large numbers of predictors
Saliency maps ("Saliency maps" section in "Interpretation methods for deep learning" section)	Results can be presented in image space (often easier for humans to examine) Example-by-example and multiexample explanations: meaningful results can be presented for single examples; can also be a disadvantage but can be alleviated by compositing Explains how neuron activation changes with each input value (i.e., each predictor at each grid point) Can be used for neurons, channels, or other groupings of neurons	Differentiable models only Simulates only slight change to input data (linear approximation to derivative)
Grad-CAM ("Gradient-weighted class-activation maps" section)	Results can be presented in image space Example-by-example and multiexample Identifies important locations in the images	Deep learning only Does not explain how values at these locations influence neuron activation
Backward optimization ("Backward optimization" section in "Interpretation methods for deep learning" section)	Results can be presented in image space Example-by-example and multiexample Extends saliency maps (uses derivatives to optimize image for desired neuron activation) Shows model behavior for extreme case	Deep learning and differentiable models only Can produce physically unrealistic output Answer depends heavily on initial seed
Novelty detection ("Novelty detection" section in "Interpretation methods for deep learning" section)	Results can be presented in image space Example-by-example and multiexample Finds interesting images and/or image subsets for further analysis	Deep learning only Depends on upconvnet, which can be difficult to train

major disadvantage: it can be used only for tree-based models. Permutation is more general (can be applied to traditional and DL models) but more computationally expensive, especially for the multipass version. Sequential selection is also general but even more computationally expensive, as it requires retraining the model potentially thousands or millions of times. Also, none of these methods explain how or why a

predictor is important, nor do they differentiate between situations where the predictor is important and is not. This question can be answered by partial-dependence plots and some of the DL-based methods.

DL-based interpretation methods can identify important spatial locations and spatial multivariate patterns, create synthetic data that minimize or maximize a certain prediction, identify novel examples in the dataset, and identify the novel parts of each example. Most DL-based methods can be applied to different parts of the model, for example, a neuron in any layer, a group of neurons in any layer, or the final prediction, which allows these methods to explain what the model "sees" at different depths (e.g., Fig. 8). Compared to the permutation test and sequential selection, most of the DL-based methods discussed are computationally efficient. This is primarily because these methods do not involve retraining the DL model. However, training the DL model in the first place can be computationally expensive.

Table 2. Mapping MIV methods to tasks.

Method	Tasks
Impurity importance ("Impurity importance" section)	Quick computation of predictor importance for tree-based methods
Permutation importance ("Permutation importance" section)	Ranks and quantifies importance of each predictor; works for any model (not only trees)
Sequential selection ["Sequential (forward and backward) selection" section]	Identify a minimal set of predictors to build a model
Partial-dependence plots ("Partial-dependence plots" section in "Interpretation and visualization methods for traditional machine learning" section)	Identify sensitivity of a model to a predictor over its full range
Saliency maps ("Saliency maps" section in "Interpretation methods for deep learning" section)	Visualize local gradient of predictions with respect to the predictors in the input space
Grad-CAM ("Gradient-weighted class-activation maps" section)	Visualize most important spatial regions of the predictor space; deep learning only
Backward optimization ("Backward optimization" section in "Interpretation methods for deep learning" section)	Create synthetic examples that activate the model in a certain way (e.g., minimize or maximize prediction); deep learning only
Novelty detection ("Novelty detection" section in "Interpretation methods for deep learning" section)	Identify novel/unexpected examples and what region of each example makes it novel; deep learning only

It is crucial to understand the trade-offs between predictability and interpretability. Since ML models are not inherently modeling a physical problem, they may find solutions with better predictive skill at the cost of a less interpretable model. For example, different ML models inherently learn different types of solutions. If one method has better predictive skill and chooses a different set of important predictors, this does not necessarily mean that the other predictors are physically unimportant. This is a common pitfall in recent MIV papers in physical science, and we caution new users to understand the limitations of and differences among MIV methods before making physical conclusions.

One aspect of the problem not discussed in this paper is formal hypothesis testing. To conclude that ML has confirmed existing knowledge or discovered something new, would require robust hypothesis testing. We are currently determining the best ways to incorporate this into our work.

In addition to explaining the behavior of the model, one potential use of MIV in meteorology is to identify new hypotheses for scientists to explore. This potential is becoming more attractive as datasets grow and prediction tasks become harder, while ML accordingly becomes more sophisticated and more integrated into our workflow. Since ML can process data quickly, it could be used to "flag" interesting data (e.g., patterns or interactions among predictors) for further analysis. This has already been done with novelty detection, which is used to flag images taken by the Mars rovers as targets for future exploration by the rovers (Wagstaff and Lee 2018). In meteorology, such methods could be used, for example, to identify observations that need to be collected more often or processes that need to be better resolved in physical models. This would allow for data science to feed back on and enrich physical science.

ACKNOWLEDGMENTS. This material is based upon work supported by the National Science Foundation under Grant EAGER AGS 1802627. 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. Most of the computing for this project was performed at the OU Supercomputing Center for Education and Research (OSCER) at the University of Oklahoma (OU). The hail research was funded through the NCAR Advanced Study Program Postdoctoral Fellowship, and computing for the project was performed on the NCAR Cheyenne supercomputer (Computational and Information Systems Laboratory 2017) and HPC Futures Lab. The National Center for Atmospheric Research is sponsored by the National Science Foundation.

REFERENCES

Adebayo, J., J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, 2018: Sanity checks for saliency maps. *Conf. on Neural Information Processing Systems*, Montreal, QC, Canada, Neural Information Processing Systems Foundation.

Benjamin, S., and Coauthors, 2004: An hourly assimilation–forecast cycle: The RUC. *Mon. Wea. Rev.*, **132**, 495–518, https://doi.org/10.1175/1520-0493(2004)132<0495:AHACTR>2.0.CO;2.

——, and Coauthors, 2016: A North American hourly assimilation and model forecast cycle: The Rapid Refresh. *Mon. Wea. Rev.*, **144**, 1669–1694, https://doi.org/10.1175/MWR-D-15-0242.1.

Bergthórsson, P., and B. Döös, 1955: Numerical weather map analysis. *Tellus*, **7**, 329–340, https://doi.org/10.3402/tellusa.v7i3.8902.

Billet, J., M. DeLisi, B. Smith, and C. Gates, 1997: Use of regression techniques to predict hail size and the probability of large hail. *Wea. Forecasting*, **12**, 154–164, https://doi.org/10.1175/1520-0434(1997)012<0154:UORTTP>2.0.CO;2.

Breiman, L., 1984: *Classification and Regression Trees*. Routledge, 358 pp.

——, 2001: Random forests. *Mach. Learn.*, **45**, 5–32, https://doi.org/10.1023/A:1010933404324.

Brotzge, J., S. Nelson, R. Thompson, and B. Smith, 2013: Tornado probability of detection and lead time as a function of convective mode and environmental parameters. *Wea. Forecasting*, **28**, 1261–1276, https://doi.org/10.1175/WAF-D-12-00119.1.

Bunkers, M., B. Klimowski, J. Zeitler, R. Thompson, and M. Weisman, 2000: Predicting supercell motion using a new hodograph technique. *Wea. Forecasting*, **15**, 61–79, https://doi.org/10.1175/1520-0434(2000)015<0061:PSMUAN>2.0.CO;2.

Carter, B., J. Mueller, S. Jain, and D. Gifford, 2018: What made you do this? Understanding black-box decisions with sufficient input subsets. arXiv, https://arxiv.org/abs/1810.03805.

Chisholm, D., J. Ball, K. Veigas, and P. Luty, 1968: The diagnosis of upper-level humidity. *J. Appl. Meteor.*, **7**, 613–619, https://doi.org/10.1175/1520-0450(1968)007<0613:TDOULH>2.0.CO;2.

Chollet, F., 2015: Keras. GitHub, https://github.com/keras-team/keras.

——, 2018: *Deep Learning with Python*. Manning, 384 pp.

Cintineo, J., M. Pavolonis, J. Sieglaff, and D. Lindsey, 2014: An empirical model for assessing the severe weather potential of developing convection. *Wea. Forecasting*, **29**, 639–653, https://doi.org/10.1175/WAF-D-13-00113.1.

——, and Coauthors, 2018: The NOAA/CIMSS ProbSevere model: Incorporation of total lightning and validation. *Wea. Forecasting*, **33**, 331–345, https://doi.org/10.1175/WAF-D-17-0099.1.

Clark, A., 2017: Generation of ensemble mean precipitation forecasts from convection-allowing ensembles. *Wea. Forecasting*, **32**, 1569–1583, https://doi.org/10.1175/WAF-D-16-0199.1.

——, A. MacKenzie, A. McGovern, V. Lakshmanan, and R. Brown, 2015: An automated, multiparameter dryline identification algorithm. *Wea. Forecasting*, **30**, 1781–1794, https://doi.org/10.1175/WAF-D-15-0070.1.

Computational and Information Systems Laboratory, 2017: Cheyenne: HPE/SGI ICE XA System (NCAR Community Computing). National Center for Atmospheric Research, https://doi.org/10.5065/d6rx99hx.

Cressman, G., 1959: An operational objective analysis system. *Mon. Wea. Rev.*, **87**, 367–374, https://doi.org/10.1175/1520-0493(1959)087<0367:AOOAS>2.0.CO;2.

Dieleman, S., K. Willett, and J. Dambre, 2015: Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Mon. Not. Roy. Astron. Soc.*, **450**, 1441–1459, https://doi.org/10.1093/mnras/stv632.

Dosovitskiy, A., and T. Brox, 2016: Inverting visual representations with convolutional networks. *Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, IEEE.

Drucker, H., C. Burges, L. Kaufman, A. Smola, and V. Vapnik, 1997: Support vector regression machines. *Conf. on Neural Information Processing Systems*, Denver, CO, Neural Information Processing Systems Foundation, 155–161.

Ebert, E., 2001: Ability of a poor man's ensemble to predict the probability and distribution of precipitation. *Mon. Wea. Rev.*, **129**, 2461–2480, https://doi.org/10.1175/1520-0493(2001)129<2461:AOAPMS>2.0.CO;2.

Elmore, K., and H. Grams, 2016: Using mPING data to generate random forests for precipitation type

forecasts. *14th Conf. on Artificial and Computational Intelligence and its Applications to the Environmental Sciences*, New Orleans, LA, Amer. Meteor. Soc., 4.2, https://ams.confex.com/ams/96Annual/webprogram/Paper289684.html.

——, Z. Flamig, V. Lakshmanan, B. Kaney, V. Farmer, H. Reeves, and L. Rothfusz, 2014: mPING: Crowd-sourcing weather reports for research. *Bull. Amer. Meteor. Soc.*, **95**, 1335–1342, https://doi.org/10.1175/BAMS-D-13-00014.1.

Franc, V., and V. Hlavac, 2002: Multi-class support vector machine. *Int. Conf. on Pattern Recognition*, Quebec City, QC, Canada, IEEE.

Friedman, J., 2001: Greedy function approximation: A gradient boosting machine. *Ann. Stat.*, **29**, 1189–1232, https://doi.org/10.1214/aos/1013203451.

——, 2002: Stochastic gradient boosting. *Comput. Stat. Data Anal.*, **38**, 367–378, https://doi.org/10.1016/S0167-9473(01)00065-2.

Fukushima, K., and S. Miyake, 1982: Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognit.*, **15**, 455–469, https://doi.org/10.1016/0031-3203(82)90024-3.

Gagne, D., A. McGovern, and J. Brotzge, 2009: Classification of convective areas using decision trees. *J. Atmos. Oceanic Technol.*, **26**, 1341–1353, https://doi.org/10.1175/2008JTECHA1205.1.

——, ——, ——, and M. Xue, 2013: Severe hail prediction within a spatiotemporal relational data mining framework. *Int. Conf. on Data Mining*, Dallas, TX, IEEE, https://doi.org/10.1109/ICDMW.2013.121.

——, ——, S. Haupt, R. Sobash, J. Williams, and M. Xue, 2017a: Storm-based probabilistic hail forecasting with machine learning applied to convection-allowing ensembles. *Wea. Forecasting*, **32**, 1819–1840, https://doi.org/10.1175/WAF-D-17-0010.1.

——, ——, ——, and J. Williams, 2017b: Evaluation of statistical learning configurations for gridded solar irradiance forecasting. *Sol. Energy*, **150**, 383–393, https://doi.org/10.1016/j.solener.2017.04.031.

——, S. Haupt, and D. Nychka, 2019: Interpretable deep learning for spatial analysis of severe hailstorms. *Mon. Wea. Rev.*, **147**, 2827–2845, https://doi.org/10.1175/MWR-D-18-0316.1.

Goldstein, A., A. Kapelner, J. Bleich, and E. Pitkin, 2015: Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *J. Comput. Graph. Stat.*, **24**, 44–65, https://doi.org/10.1080/10618600.2014.907095.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, 2014: Generative adversarial nets. *Conf. on Neural Information Processing Systems*, Montreal, QC, Canada, Neural Information Processing Systems Foundation.

Haykin, S., 2001: Feedforward neural networks: An introduction. *Nonlinear Dynamical Systems: Feedforward Neural Network Perspectives*, I. Sandberg, Ed., John Wiley and Sons, 1–16.

Homeyer, C., and K. Bowman, 2017: Algorithm description document for version 3.1 of the three-dimensional gridded NEXRAD WSR-88D radar (GridRad) dataset. University of Oklahoma Tech. Rep., 23 pp., http://gridrad.org/pdf/GridRad-v3.1-Algorithm-Description.pdf.

Jergensen, G., 2019: PermutationImportance. GitHub, https://github.com/gelijergensen/PermutationImportance.

——, A. McGovern, R. Lagerquist, and T. Smith, 2019: Classifying convective storms using machine learning. *Wea. Forecasting*, in press.

Johns, E., O. Aodha, and G. Brostow, 2015: Becoming the expert: Interactive multi-class machine teaching. *Conf. on Computer Vision and Pattern Recognition*, Boston, MA, IEEE.

Kitzmiller, D., W. McGovern, and R. Saffle, 1995: The WSR-88D severe weather potential algorithm. *Wea. Forecasting*, **10**, 141–159, https://doi.org/10.1175/1520-0434(1995)010<0141:TWSWPA>2.0.CO;2.

Kohavi, R., and G. John, 1997: Wrappers for feature subset selection. *Artif. Intell.*, **97**, 273–324, https://doi.org/10.1016/S0004-3702(97)00043-X.

Kumjian, M., and A. Ryzhkov, 2008: Polarimetric signatures in supercell thunderstorms. *J. Appl. Meteor. Climatol.*, **47**, 1940–1961, https://doi.org/10.1175/2007JAMC1874.1.

Kunkel, K., J. Biard, and E. Racah, 2018: Automated detection of fronts using a deep learning algorithm. *17th Conf. on Artificial and Computational Intelligence and its Applications to the Environmental Sciences*, Austin, TX, Amer. Meteor. Soc., TJ7.4, https://ams.confex.com/ams/98Annual/webprogram/Paper333480.html.

Lagerquist, R., 2018: Deep learning for prediction of meteorological fronts: Keras tutorial. GitHub, https://github.com/thunderhoser/aiml_symposium/blob/master/aiml_symposium/aiml_symposium.ipynb.

——, and D. Gagne II, 2019: Interpretation of deep-learning models for predicting thunderstorm rotation: Python tutorial. GitHub, https://github.com/djgagne/ams-ml-python-course/blob/ryan_branch/module_4/ML_Short_Course_Module_4_Interpretation.ipynb.

——, A. McGovern, and T. Smith, 2017: Machine learning for real-time prediction of damaging straight-line

convective wind. *Wea. Forecasting*, **32**, 2175–2193, https://doi.org/10.1175/WAF-D-17-0038.1.

——, C. Homeyer, A. McGovern, C. Potvin, T. Sandmael, and T. Smith, 2018: Deep learning for real-time storm-based tornado prediction. *29th Conf. on Severe Local Storms*, Stowe, VT, Amer. Meteor. Soc., 138, https://ams.confex.com/ams/29SLS/webprogram/Paper348817.html.

——, ——, ——, ——, ——, and ——, 2019a: Development and interpretation of deep-learning models for nowcasting convective hazards. *18th Conf. on Artificial and Computational Intelligence and its Applications to the Environmental Sciences*, Phoenix, AZ, Amer. Meteor. Soc., 3B.1, https://ams.confex.com/ams/2019Annual/meetingapp.cgi/Paper/352846.

——, A. McGovern, and D. Gagne II, 2019b: Deep learning for spatially explicit prediction of synoptic-scale fronts. *Wea. Forecasting*, **34**, 1137–1160, https://doi.org/10.1175/WAF-D-18-0183.1.

Lakshmanan, V., C. Karstens, J. Krause, K. Elmore, A. Ryzhkov, and S. Berkseth, 2015: Which polarimetric variables are important for weather/no-weather discrimination? *J. Atmos. Oceanic Technol.*, **32**, 1209–1223, https://doi.org/10.1175/JTECH-D-13-00205.1.

Leardi, R., 1996: Genetic algorithms in feature selection. *Genetic Algorithms in Molecular Modeling*, J. Devillers, Ed., Academic Press, 67–86.

LeCun, Y., Y. Bengio, and G. Hinton, 2015: Deep learning. *Nature*, **521**, 436–444, https://doi.org/10.1038/nature14539.

Lipton, Z., 2016: The mythos of model interpretability. *Int. Conf. on Machine Learning: Workshop on Human Interpretability in Machine Learning*, New York, NY, International Machine Learning Society.

Liu, Y., and Coauthors, 2016: Application of deep convolutional neural networks for detecting extreme weather in climate datasets. arXiv, https://arxiv.org/abs/1605.01156.

Louppe, G., L. Wehenkel, A. Sutera, and P. Geurts, 2013: Understanding variable importances in forests of randomized trees. *Conf. on Neural Information Processing Systems*, Lake Tahoe, CA, Neural Information Processing Systems Foundation.

Maas, A., A. Hannun, and A. Ng, 2013: Rectifier nonlinearities improve neural network acoustic models. *Int. Conf. on Machine Learning*, Atlanta, GA, International Machine Learning Society.

Mahesh, A., T. O'Brien, M. Prabhat, and W. Collins, 2018: Assessing uncertainty in deep learning techniques that identify atmospheric rivers in climate simulations. *17th Conf. on Artificial and Computational Intelligence and its Applications to the Environmental Sciences*, Austin, TX, Amer. Meteor. Soc., 2.2, https://ams.confex.com/ams/pdfpapers/138754.pdf.

Malone, T., 1955: Application of statistical methods in weather prediction. *Proc. Natl. Acad. Sci. USA*, **41**, 806–815, https://doi.org/10.1073/pnas.41.11.806.

Markowski, P., J. Straka, and E. Rasmussen, 2002: Direct surface thermodynamic observations within the rear-flank downdrafts of nontornadic and tornadic supercells. *Mon. Wea. Rev.*, **130**, 1692–1721, https://doi.org/10.1175/1520-0493(2002)130<1692:DSTOWT>2.0.CO;2.

Marzban, C., and G. Stumpf, 1996: A neural network for tornado prediction based on Doppler radar-derived attributes. *J. Appl. Meteor.*, **35**, 617–626, https://doi.org/10.1175/1520-0450(1996)035<0617:ANNFTP>2.0.CO;2.

——, and ——, 1998: A neural network for damaging wind prediction. *Wea. Forecasting*, **13**, 151–163, https://doi.org/10.1175/1520-0434(1998)013<0151:ANNFDW>2.0.CO;2.

——, and A. Witt, 2001: A Bayesian neural network for severe-hail size prediction. *Wea. Forecasting*, **16**, 600–610, https://doi.org/10.1175/1520-0434(2001)016<0600:ABNNFS>2.0.CO;2.

McGovern, A., D. Gagne II, J. Williams, R. Brown, and J. Basara, 2014: Enhancing understanding and improving prediction of severe weather through spatiotemporal relational learning. *Mach. Learn.*, **95**, 27–50, https://doi.org/10.1007/s10994-013-5343-x.

——, ——, J. Basara, T. Hamill, and D. Margolin, 2015: Solar energy prediction: An international contest to initiate interdisciplinary research on compelling meteorological problems. *Bull. Amer. Meteor. Soc.*, **96**, 1388–1395, https://doi.org/10.1175/BAMS-D-14-00006.1.

——, K. Elmore, D. Gagne, S. Haupt, C. Karstens, R. Lagerquist, T. Smith, and J. Williams, 2017: Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bull. Amer. Meteor. Soc.*, **98**, 2073–2090, https://doi.org/10.1175/BAMS-D-16-0123.1.

——, G. Jergensen, C. Karstens, H. Obermeier, and T. Smith, 2018: Real-time and climatological storm classification using machine learning. *17th Conf. on Artificial and Computational Intelligence and its Applications to the Environmental Sciences*, Austin, TX, Amer. Meteor. Soc., 1.1, https://ams.confex.com/ams/98Annual/webprogram/Paper326198.html.

Metz, C., 1978: Basic principles of ROC analysis. *Semin. Nucl. Med.*, **8**, 283–298, https://doi.org/10.1016/S0001-2998(78)80014-2.

Mitchell, T., 1997: *Machine Learning*. McGraw Hill, 414 pp.

Molnar, C., 2018: *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Leanpub, https://christophm.github.io/interpretable-ml-book/.

Montavon, G., W. Samek, and K. Müller, 2018: Methods for interpreting and understanding deep neural networks. *Digital Signal Process.*, **73**, 1–15, https://doi.org/10.1016/j.dsp.2017.10.011.

Nair, V., and G. Hinton, 2010: Rectified linear units improve restricted Boltzmann machines. *Proc. 27th Int. Conf. on Machine Learning*, Haifa, Israel, International Machine Learning Society, 807–814, https://dl.acm.org/citation.cfm?id=3104425.

National Weather Service, 2016: *Storm Data* preparation. National Weather Service Instruction 10-1605, 110 pp., www.nws.noaa.gov/directives.

NeurIPS Foundation, 2018: NeurIPS Workshop on Interpretability and Robustness in Audio, Speech, and Language. GitHub, https://nips.cc/Conferences/2018/Schedule?showEvent=10917.

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

Ortega, K., T. Smith, J. Zhang, C. Langston, Y. Qi, S. Stevens, and J. Tate, 2012: The Multi-Year Reanalysis of Remotely Sensed Storms (MYRORSS) project. *26th Conf. on Severe Local Storms*, Nashville, TN, Amer. Meteor. Soc., 74, https://ams.confex.com/ams/26SLS/webprogram/Paper211413.html.

Panofsky, R., 1949: Objective weather-map analysis. *J. Meteor.*, **6**, 386–392, https://doi.org/10.1175/1520-0469(1949)006<0386:OWMA>2.0.CO;2.

Pedregosa, F., and Coauthors, 2011: Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.

Provost, F., and P. Domingos, 2003: Tree induction for probability-based ranking. *Mach. Learn.*, **52**, 199–215, https://doi.org/10.1023/A:1024099825458.

Quinlan, J., 1986: Induction of decision trees. *Mach. Learn.*, **1**, 81–106.

Radhika, Y., and M. Shashi, 2009: Atmospheric temperature prediction using support vector machines. *Int. J. Comput. Theory Eng.*, **1**, 55–58, https://doi.org/10.7763/IJCTE.2009.V1.9.

Rajasekhar, N., and T. Rajinikanth, 2014: Hybrid SVM data-mining techniques for weather data analysis of Krishna district of Andhra region. *Int. J. Res. Comput. Commun. Technol.*, **3**, 743–748.

Rakhlin, A., A. Shvets, V. Iglovikov, and A. Kalinin, 2018: Deep convolutional neural networks for breast cancer histology image analysis. arXiv, https://arxiv.org/abs/1802.00752.

Rao, T., N. Rajasekhar, and T. Rajinikanth, 2012: An efficient approach for weather forecasting using support vector machines. *Int. Conf. on Computer Technology and Science*, New Delhi, India, International Association of Computer Science and Information Technology.

Ribeiro, M., S. Singh, and C. Guestrin, 2016: "Why should I trust you?": Explaining the predictions of any classifier. *Int. Conf. on Knowledge Discovery and Data Mining*, San Francisco, CA, Association for Computing Machinery, https://doi.org/10.1145/2939672.2939778.

Richman, M., 1986: Rotation of principal components. *J. Climatol.*, **6**, 293–335, https://doi.org/10.1002/joc.3370060305.

Samek, W., T. Wiegand, and K. Müller, 2017: Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv, https://arxiv.org/abs/1708.08296.

Sandmael, T., and C. Homeyer, 2018: Comparison of tornadic and severe non-tornadic storms using probability matched means of radar observations. *29th Conf. on Severe Local Storms*, Stowe, VT, Amer. Meteor. Soc., 82, https://ams.confex.com/ams/29SLS/webprogram/Paper348255.html.

Schölkopf, B., K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, 1997: Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Trans. Signal Proc.*, **45**, 2758–2765, https://doi.org/10.1109/78.650102.

Schwartz, C., G. Romine, M. Weisman, R. Sobash, K. Fossell, K. Manning, and S. Trier, 2015: A real-time convection-allowing ensemble prediction system initialized by mesoscale ensemble Kalman filter analyses. *Wea. Forecasting*, **30**, 1158–1181, https://doi.org/10.1175/WAF-D-15-0013.1.

Selvaraju, R., M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, 2017: Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Int. Conf. on Computer Vision*, Venice, Italy, IEEE, https://doi.org/10.1109/ICCV.2017.74.

Siedlecki, W., and J. Sklansky, 1993: A note on genetic algorithms for large-scale feature selection. *Handbook of Pattern Recognition and Computer Vision*, C. Chen, L. Pau, and P. Wang, Eds., World Scientific, 88–107.

Silver, D., and Coauthors, 2016: Mastering the game of go with deep neural networks and tree search. *Nature*, **529**, 484–489, https://doi.org/10.1038/nature16961.

——, and Coauthors, 2017: Mastering the game of go without human knowledge. *Nature*, **550**, 354–359, https://doi.org/10.1038/nature24270.

Simonyan, K., A. Vedaldi, and A. Zisserman, 2014: Deep inside convolutional networks: Visualizing image classification models and saliency maps. arXiv, https://arxiv.org/abs/1312.6034.

Smith, B., R. Thompson, J. Grams, C. Broyles, and H. Brooks, 2012: Convective modes for significant severe thunderstorms in the contiguous United States. Part I: Storm classification and climatology. *Wea. Forecasting*, **27**, 1114–1135, https://doi.org/10.1175/WAF-D-11-00115.1.

Stracuzzi, D., and P. Utgoff, 2004: Randomized variable elimination. *J. Mach. Learn. Res.*, **5**, 1331–1362.

Thompson, G., R. Rasmussen, and K. Manning, 2004: Explicit forecasts of winter precipitation using an improved bulk microphysics scheme. Part I: Description and sensitivity analysis. *Mon. Wea. Rev.*, **132**, 519–542, https://doi.org/10.1175/1520-0493(2004)132<0519:EFOWPU>2.0.CO;2.

——, P. Field, R. Rasmussen, and W. Hall, 2008: Explicit forecasts of winter precipitation using an improved bulk microphysics scheme. Part II: Implementation of a new snow parameterization. *Mon. Wea. Rev.*, **136**, 5095–5115, https://doi.org/10.1175/2008MWR2387.1.

Thompson, R., B. Smith, J. Grams, A. Dean, and C. Broyles, 2012: Convective modes for significant severe thunderstorms in the contiguous United States. Part II: Supercell and QLCS tornado environments. *Wea. Forecasting*, **27**, 1136–1154, https://doi.org/10.1175/WAF-D-11-00116.1.

Vapnik, V., 1963: Pattern recognition using generalized portrait method. *Autom. Remote Control*, **24**, 774–780.

——, 1995: *The Nature of Statistical Learning Theory.* Springer, 188 pp.

Wade, A., M. Coniglio, and C. Ziegler, 2018: Comparison of near- and far-field supercell inflow environments using radiosonde observations. *Mon. Wea. Rev.*, **146**, 2403–2415, https://doi.org/10.1175/MWR-D-17-0276.1.

Wagstaff, K., and J. Lee, 2018: Interpretable discovery in large image data sets. arXiv, https://arxiv.org/abs/1806.08340.

Webb, A., 2003: *Statistical Pattern Recognition.* John Wiley and Sons, 514 pp.

Wilks, D., 2006: *Statistical Methods in the Atmospheric Sciences.* 2nd ed. Academic Press, 627 pp.

Williams, J., 2014: Using random forests to diagnose aviation turbulence. *Mach. Learn.*, **95**, 51–70, https://doi.org/10.1007/s10994-013-5346-7.

——, D. Ahijevych, S. Dettling, and M. Steiner, 2008a: Combining observations and model data for short-term storm forecasting. *Proc. SPIE*, **7088**, 708805, https://doi.org/10.1117/12.795737.

——, R. Sharman, J. Craig, and G. Blackburn, 2008b: Remote detection and diagnosis of thunderstorm turbulence. *Proc. SPIE*, **7088**, 708804, https://doi.org/10.1117/12.795570.

Zhou, B., A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, 2016: Learning deep features for discriminative localization. *Conf. on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, IEEE, https://doi.org/10.1109/CVPR.2016.319.

Attention AMS Student Members

Stay connected to AMS after graduation for half the regular membership rate

AMS
American Meteorological Society

Let AMS help you build your expertise, your network, your career. There's never been a more important time to be a member.

http://www.ametsoc.org/earlycareer