Deep Learning for Spatially Explicit Prediction of Synoptic-Scale Fronts

RYAN LAGERQUIST

Cooperative Institute for Mesoscale Meteorological Studies, and University of Oklahoma, Norman, Oklahoma

AMY MCGOVERN

University of Oklahoma, Norman, Oklahoma

DAVID JOHN GAGNE II

National Center for Atmospheric Research, Boulder, Colorado

(Manuscript received 31 October 2018, in final form 7 June 2019)

ABSTRACT

This paper describes the use of convolutional neural nets (CNN), a type of deep learning, to identify fronts in gridded data, followed by a novel postprocessing method that converts probability grids to objects. Synoptic-scale fronts are often associated with extreme weather in the midlatitudes. Predictors are 1000-mb (1 mb = 1 hPa) grids of wind velocity, temperature, specific humidity, wet-bulb potential temperature, and/or geopotential height from the North American Regional Reanalysis. Labels are human-drawn fronts from Weather Prediction Center bulletins. We present two experiments to optimize parameters of the CNN and object conversion. To evaluate our system, we compare the objects (predicted warm and cold fronts) with human-analyzed warm and cold fronts, matching fronts of the same type within a 100- or 250-km neighborhood distance. At 250 km our system obtains a probability of detection of 0.73, success ratio of 0.65 (or false-alarm rate of 0.35), and critical success index of 0.52. These values drastically outperform the baseline, which is a traditional method from numerical frontal analysis. Our system is not intended to replace human meteorologists, but to provide an objective method that can be applied consistently and easily to a large number of cases. Our system could be used, for example, to create climatologies and quantify the spread in forecast frontal properties across members of a numerical weather prediction ensemble.

1. Introduction

Synoptic-scale fronts are often associated with extreme weather in the midlatitudes (e.g., Fawbush and Miller 1954; Miller 1959; Catto and Pfahl 2013). A front is a quasi-vertical transition zone between two air masses with different densities (American Meteorological Society 2014b). This definition assumes that the two air masses are nearly horizontally homogeneous internally, with a sharp density gradient in the transition zone. It is common practice to think of fronts as infinitesimally thin horizontal lines, located at the warm edge of the transition zone (American Meteorological Society 2014b). For synoptic-scale fronts, the width of this transition zone is on the order of 100 km or less (section 9.2, Holton 2004). Thus, "synoptic-scale" fronts are mesoscale phenomena in the cross-front direction, despite being synoptic scale (from a few hundred to a few thousand kilometers) in the alongfront direction. Also, though the fundamental definition involves density, fronts are commonly defined by a thermal variable such as (potential) temperature, wet-bulb (potential) temperature, or equivalent (potential) temperature, as discussed later in this section.

Air masses are generated by prolonged residence in a source region, where they are conditioned (thermally altered) by contact with the underlying surface. For example, around the North Pole during winter, the atmosphere receives no solar radiation and continually emits longwave radiation to space, leading to strong cooling (Serreze et al. 2007). Meanwhile, over the equatorial oceans, the atmosphere receives ample solar radiation, as well as longwave radiation and water vapor

Supplemental information related to this paper is available at the Journals Online website: https://doi.org/10.1175/WAF-D-18-0183.s1.

Corresponding author: Ryan Lagerquist, ryan.lagerquist@ou.edu

DOI: 10.1175/WAF-D-18-0183.1

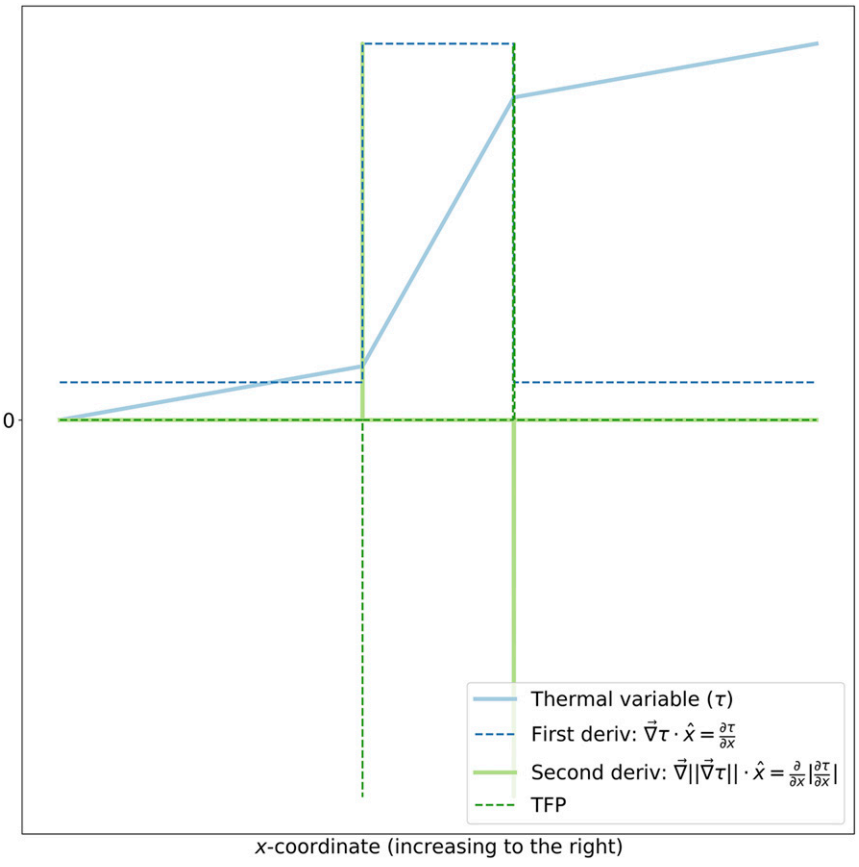

FIG. 1. Schematic for TFP calculation [Eq. (1)]. The variable τ increases gradually with x, except in the middle of the domain (frontal zone), where it increases sharply with x. Both the first and second derivatives have their greatest magnitude at the edges of the frontal zone, so TFP has its greatest magnitude at the edges of the frontal zone. At the warm edge, the first and second derivatives have opposite sign, so TFP reaches its most positive value here. At the cold edge, the first and second derivatives have the same sign, so TFP reaches its most negative value here.

from the underlying surface. These processes lead to continental Arctic and maritime tropical air masses, respectively. Air masses are often characterized by their wet-bulb potential temperature[1] θ_w (Hewson 1936; Low and Hudak 1997). Wet-bulb potential temperature θ_w is a good discriminator because it is conservative under psuedo-adiabatic processes, which occur frequently within air masses.

Numerical frontal analysis (NFA) is the processing of grids without machine learning and dates back to Renard and Clarke (1965, hereafter RC65), who define the thermal front parameter [TFP; Eq. (1)]. The term τ may be any thermal variable [(potential) temperature, wet-bulb (potential) temperature, equivalent (potential) temperature, etc.] at one vertical level or averaged over a layer. Here, $\hat{\nabla}\tau = \nabla\tau/\nabla\tau$ is the unit vector in the direction of the gradient:

$$\mathrm{TFP}(\tau) = -\nabla||\nabla\tau|| \cdot \hat{\nabla}\tau. \quad (1)$$

Ridge lines of TFP (maxima) define the warm edge of the frontal zone, and trough lines (minima) define the cold edge, as shown in Fig. 1. RC65 draw these lines manually, ignoring small regions of enhanced TFP (in order to highlight the synoptic scale). Their thermal variable τ is θ_{850}.

Clarke and Renard (1966, hereafter CR66) introduce the first thermal front locator, defined in Eq. (2):

$$\mathrm{TFL}_1(\tau) = \nabla\tau \cdot \hat{\nabla}\mathrm{TFP}(\tau). \quad (2)$$

TFL_1 is the thermal gradient in the direction of the TFP gradient, from the cold side toward the warm side of the

[1] This and other atmospheric variables are defined in Table 1. Henceforth, if we use a mathematical variable without definition in the main text, it is defined in Table 1.

TABLE 1. Glossary.

Term	Units	Definition
NF	—	No-front label. A grid cell labeled NF is not intersected by a front.
WF	—	Warm front label. A grid cell labeled WF is intersected by a warm front.
CF	—	Cold front label. A grid cell labeled CF is intersected by a cold front.
T	kelvins (K)	Temperature
q	kg kg^{-1}	Specific humidity
u	m s^{-1}	Zonal wind speed
v	m s^{-1}	Meridional wind speed
Z	meters above sea level (m MSL)	Geopotential height
p	Pa	Pressure
θ	K	Potential temperature
θ_w	K	Wet-bulb potential temperature. This is the temperature that an air parcel would reach if it were brought adiabatically to saturation (e.g., by lifting to its condensation level), then adiabatically compressed or expanded to 1000 mb (American Meteorological Society 2014c).
T_{1000}, q_{850}, etc.	—	The subscript is pressure level in millibars (mb).
$\overline{T}_{1000-700}$, $\overline{q}_{850-500}$, etc.	—	Mean value between two pressure levels, given in the subscript in millibars.
T_e	K	Adiabatic equivalent temperature (American Meteorological Society 2014a)
$\Delta Z(T_e)$	m	Equivalent thickness. Same as thickness, except that in the hypsometric equation (American Meteorological Society 2014d) virtual temperature is replaced by T_e, which is conserved during psuedo-adiabatic motion within the layer.

frontal zone. The zero contour of TFL_1 defines the edges of the frontal zone. Their thermal variable τ is T_{850}.

Huber-Pock and Kress (1981, hereafter HPK81) introduce the second thermal front locator (TFL_2), defined in Eq. (3):

$$TFL_2(\tau) = \nabla TFP(\tau) \cdot \hat{\nabla}\tau. \quad (3)$$

TFL_2 is the TFP gradient in the direction of the thermal gradient, whereas TFL_1 is the thermal gradient in the direction of the TFP gradient. Again, the zero contour defines the edges of the frontal zone. HPK81 use a very different thermal variable than earlier studies: $\Delta Z(T_e)$, defined in Table 1. Also, they impose the criterion that the magnitude of the cross-front thermal gradient must exceed that of the alongfront gradient.

Hewson (1998) summarizes and codifies a lot of earlier work in NFA, including RC65, CR66, and HPK81. Specifically, they distill the process into the application of a single locating variable (e.g., the TFP, TFL_1, or TFL_2) and one or more masking variables. The ''masking variables'' are simply additional criteria, such as minimum frontal-zone area, minimum front length, minimum $||\nabla\tau||$ or TFP, etc. The generic process is shown in their Fig. 2. This framework has been used in most studies of NFA (Table 2).

Simmonds et al. (2012) develop the ''wind-shift method,'' which they find to match human analyses better than the ''thermal method'' (Hewson 1998) in the Southern Hemisphere. Specifically, the wind-shift method classifies a grid cell as frontal if, within the last six hours, 1) its wind direction has changed from the northwest quadrant to the southwest quadrant and 2) its meridional wind component v has increased by >2 m s^{-1}. These criteria are applied to wind fields at 10 m above ground or 850 mb (1 mb = 1 hPa). Schemm et al. (2015) compare the thermal and wind-shift methods, concluding that although the wind-shift method is better at detecting fronts with weak baroclinicity (e.g., those induced by wind shear and convergence between two anticyclones), the thermal method is better at detecting warm fronts (which are almost never detected by the wind-shift method).

Machine learning (ML) is a process whereby computers learn autonomously from data, as opposed to an expert system like NFA, which is based on human-derived rules. Deep learning (DL) is a subset of ML, which offers the ability to encode the input data at various levels of abstraction. These abstractions are called features, and DL autonomously learns the best way to encode features (that which maximizes predictive skill).

Convolutional neural networks (CNN), a common type of DL model, are specially designed to learn from data with topological structure, such as spatial grids. Although they were introduced in the early 1980s (Fukushima and Miyake 1982), CNNs (and DL in general) remained obscure until just a few years ago. One reason is that until recently most people did not have the computing power needed to train DL models. Second, DL models often have very many weights (millions or tens of millions; e.g., Krizhevsky et al. 2012, Chollet 2017),

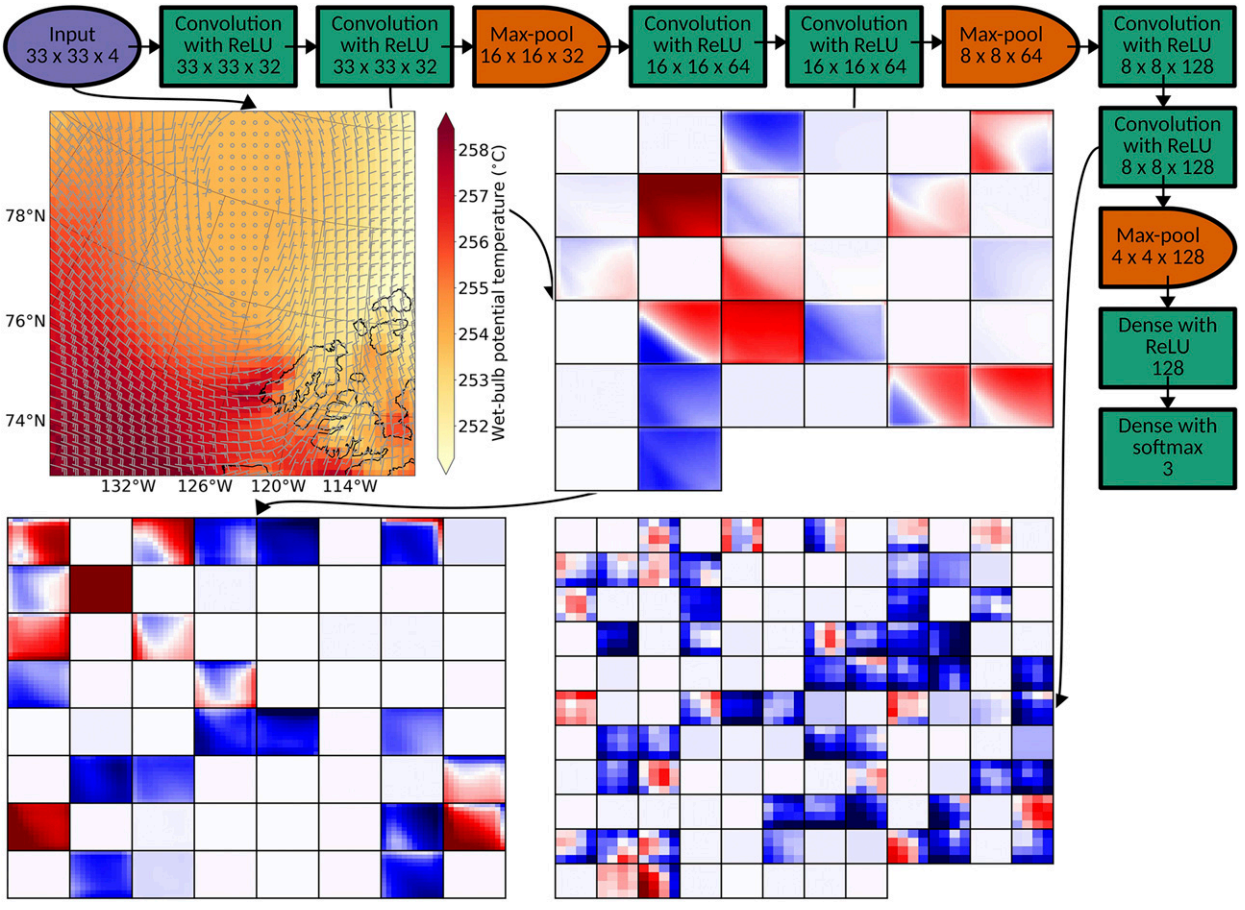

FIG. 2. Architecture of a CNN. The input (at top left) is a 33 × 33 grid of 4 variables (1000-mb temperature, specific humidity, u wind, and v wind). Wind barbs are shown in gray, and θ_w (which is a function of both temperature and humidity) is shown in the yellow-to-red fill. In the feature maps produced by convolution and pooling layers, negative values are in blue and positive values are in red. The first convolution layer transforms the 4 variables into 32 filters; the second convolution layer transforms these into 32 new filters; and the first pooling layer downsamples the 32 feature maps to half-resolution. These three layers form a "convolution block," and the next two convolution blocks perform similar operations. Feature maps from the last pooling layer (4 rows × 4 columns × 128 filters) are flattened into a vector of length 2048. These 2048 features are passed through two dense layers, which transform them into 128 intermediate features and then 3 predictions (probabilities of no front, warm front, and cold front).

which makes overfitting likely without a very large amount of training data. This problem has been alleviated by the availability of more data and dropout regularization (Hinton et al. 2012; Baldi and Sadowski 2013). Third, neural networks (including CNNs) often have many layers, which leads to the vanishing-gradient problem. Specifically, training a neural network involves computing the gradient of each weight with respect to the error [backpropagation; section 4.5.2 in Mitchell (1997)]. Most gradients are the product of other gradients, which tend to be small ($\ll$1), and the number of gradients in the product increases with the number of layers. When enough small values are multiplied together, the product can become indistinguishable from zero (numerical underflow), causing the gradient to "vanish". This problem has been alleviated by the rectified linear activation function (Nair and Hinton 2010) and batch normalization (Ioffe and Szegedy 2015).

CNNs have recently been applied to meteorology problems such as estimating sea ice concentration (Wang et al. 2016), detecting extreme weather in climate models (Racah et al. 2017; Kurth et al. 2018), detecting synoptic-scale fronts in weather models (Kunkel et al. 2018), approximating an entire global circulation model (Scher 2018), estimating tropical cyclone intensity (Wimmers et al. 2019), replacing subgrid-scale parameterizations in a climate model (Rasp et al. 2018), and forecasting tornadogenesis (McGovern et al. 2019, manuscript submitted to *Bull. Amer. Meteor. Soc.*) and large hail (Gagne et al. 2019). Also, Reichstein et al. (2019) and Gil et al. (2019) have

TABLE 2. Previous work in numerical frontal analysis. TFP is the thermal front parameter [Eq. (1)]; TFL_1 and TFL_2 are thermal front locators [Eqs. (2) and (3)]; and subscripts on thermal variables are pressure levels in millibars (except "10 m", which is 10 m above ground).

Reference	Locating variable	Domain	Grid spacing
Renard and Clarke (1965)	$TFP(\theta_{850})$	Most of Northern Hemisphere	381 km
Clarke and Renard (1966)	$TFL_1(T_{850})$, $TFP(\theta_{850})$, $TFP(\theta_{1000})$, $TFP(\overline{\theta}_{1000-700})$	Most of Northern Hemisphere	381 km
Huber-Pock and Kress (1981)	$TFL_2[\Delta Z(T_e)]$	Unknown	Unknown
Serreze et al. (2001)	$TFP(T_{850})$	North of 30°N	2.5°
Jenkner et al. (2010)	$TFP(\theta_{e700})$	South-central Europe	7 km
Hewson (1998)	Many	North Atlantic, western Europe, extreme eastern North America	~100 km
Berry et al. (2011)	$TFP(\theta_{\omega_{850}})$	Global	2.5°
Simmonds et al. (2012)	$\mathbf{v}_{10m}$, $\mathbf{v}_{850}$	Most of Southern Hemisphere	1.5°
Catto and Pfahl (2013)	$TFP(\theta_{w_{850}})$	60°S–60°N	2.5°
Schemm et al. (2015)	$TFP(\theta_{e_{850}})$, $\mathbf{v}_{10m}$	Global	1°

recently called for a vast expansion of our efforts to incorporate deep learning into geoscience.

Section 2 describes CNNs, our chosen DL model; section 3 describes our input data and preprocessing; section 4 describes postprocessing and evaluation of the CNN predictions; section 5 describes our experimental setup; and section 6 discusses the results.

2. Convolutional neural networks

The main components of a CNN (Fig. 2) are convolutional, pooling, and dense layers. Each convolutional layer passes many convolutional filters (Fig. 3) over the input maps, producing one output map for each filter. The input and output maps—more generally, spatial grids at any layer in a CNN—are called feature maps. The output maps are then passed through an activation function, which must be nonlinear. Otherwise, the network would learn only linear relationships, because convolution is a linear operation and any series of linear operations is still linear. Convolution is formulated precisely by Eq. (4):

$$\mathbf{X}_i^{(k)} = f\left\{\sum_{j=1}^{J} \mathbf{W}_i^{(j,k)} * \mathbf{X}_{i-1}^{(j)} + b_i^{(k)}\right\}, \tag{4}$$

where $\mathbf{X}_{i-1}^{(j)}$ is the jth feature map in the $(i-1)$th layer; $\mathbf{X}_i^{(k)}$ is the kth feature map in the ith layer; $\mathbf{W}_i^{(j,k)}$ is the convolutional filter connecting $\mathbf{X}_{i-1}^{(j)}$ and $\mathbf{X}_i^{(k)}$; J is the number of feature maps in the $(i-1)$th layer; $b_i^{(k)}$ is the bias for the kth feature map in the ith layer; and f is the activation function. The terms $\mathbf{X}_{i-1}^{(j)}$ and $\mathbf{X}_i^{(k)}$ are matrices with the same dimensions (e.g., 33×33 for the first convolutional layer in Fig. 2), while $\mathbf{W}_i^{(j,k)}$ typically has smaller dimensions (Fig. 3). The activation function acts independently on each element of the matrix. We use the rectified linear unit (ReLU; Nair and Hinton 2010), which is a common activation function in the ML literature.

All convolutional filters in the network have different weights ($\mathbf{W}_i^{(j,k)}$ and $b_i^{(k)}$), which is ensured by random initialization. This allows different filters to detect different features. As shown in Fig. 2, some filters respond strongly to thermal gradients, while others respond strongly to wind shifts. In reality, since convolution is performed over all input channels, the features detected are multivariate. For example, the first convolution layer convolves over the four original variables—temperature, specific humidity, u wind, and v wind—while the second convolves over the 32 channels produced by the first. During training, the weights are updated by gradient descent [Eq. (5)] to minimize the error or "loss" ϵ. Here ϵ is computed for the last batch of examples presented to the model, and α is the learning rate:

$$\begin{cases} \mathbf{W}_i^{(j,k)} \leftarrow \mathbf{W}_i^{(j,k)} - \alpha \dfrac{\partial \epsilon}{\partial \mathbf{W}_i^{(j,k)}} \\ b_i^{(k)} \leftarrow b_i^{(k)} - \alpha \dfrac{\partial \epsilon}{\partial b_i^{(k)}} \end{cases}. \tag{5}$$

Our loss is cross entropy [Eq. (6)], which is common for classification tasks. Here, N is the number of examples; K is the number of classes; p_{ik} is the predicted probability that the ith example belongs to the kth class; and y_{ik} is the true label, which is 1 if the ith example belongs to the kth class and 0 otherwise. Cross entropy varies across $[0, \infty)$; lower is better:

$$\epsilon = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\log_2(p_{ik}). \tag{6}$$

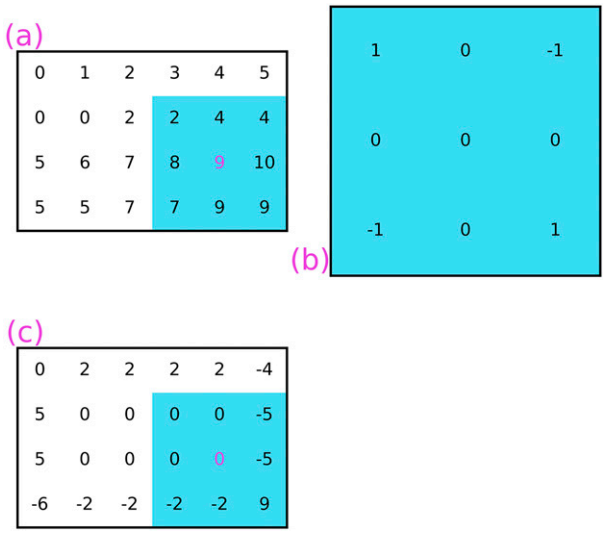

FIG. 3. Convolution. (a) Input map, with position of the filter highlighted in blue. (b) The convolutional filter. (c) Output map, with the same filter position highlighted in blue. Convolution is elementwise multiplication of the highlighted values in (a) with those in the filter, yielding the pink "0" in (c). Inputs [the numbers in (a) and (b)] were chosen arbitrarily. For the sake of simplicity, this example assumes one input map, one output map, and no activation function. If the activation function were ReLU, negative values in the output map in (c) would be set to zero, while nonnegative values would be unchanged. (This figure is part of an animation shown in Fig. S1 in the online supplemental material.)

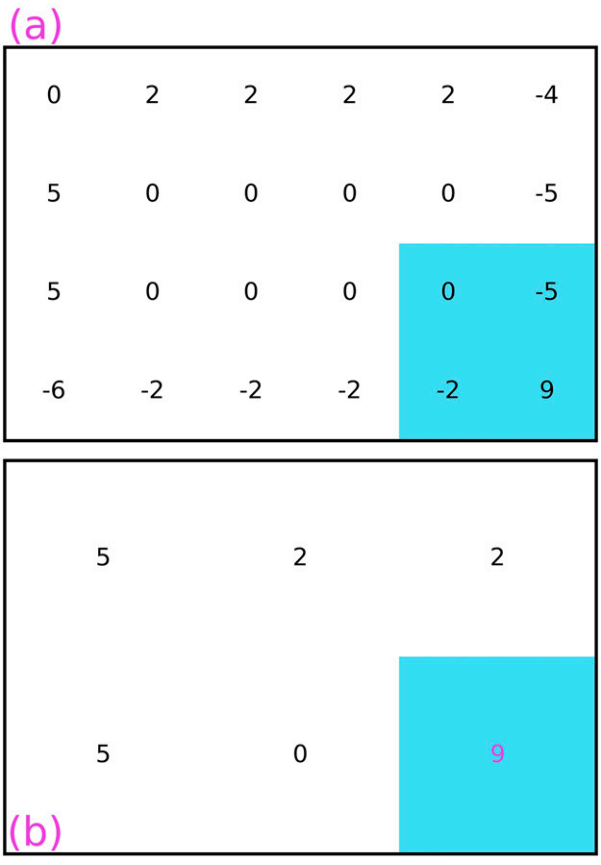

FIG. 4. Maximum pooling. (a) Input map, with position of the pooling window highlighted in blue. (b) Output map, with the resulting value highlighted in blue. This value is obtained by taking the maximum of highlighted values in (a). The other option is average pooling, which would cause the top row of the output map to become (1.75, 1, −1.75); the bottom row would be (−0.75, −1, 0.5). (This figure is part of an animation shown in Fig. S2.)

Each pooling layer moves a window over the input map and, at each position of the window, takes the maximum or mean inside the window (Fig. 4). This reduces the image size (number of grid cells) without changing the image domain; in other words, pooling reduces the image resolution (generally by half). For example, inputs to the first pooling layer in Fig. 2 have 32-km grid cells (same as the NARR), while outputs have 64-km grid cells. Pooling layers do not change the number of channels (e.g., the first pooling layer in Fig. 2 takes in 32 channels and outputs 32 channels). Reducing the image resolution allows deeper convolution layers (farther to the right in Fig. 2) to learn larger-scale features. This is one reason that deeper layers learn higher-level abstractions; the other is that feature maps at deeper layers have passed through more nonlinear transformations.

Finally, the dense layers transform feature maps into predictions. Since dense layers ignore spatial structure, before passing feature maps to the dense layers, it is common practice to flatten them into a vector—as in Fig. 2, where the $4 \times 4 \times 128$ grid is flattened to a vector of length 2048. A traditional neural net (TNN; Haykin 2001) consists of many dense layers (called "hidden layers" in the TNN literature), with no convolution or pooling layers. Thus, including dense layers in a CNN is equivalent to appending a TNN after the convolution and pooling layers. These spatially aware layers detect important spatial features, and the dense layers transform these features into predictions.

To train a TNN with spatial grids, the scalar features must be decided a priori. Some examples are raw gridpoint values (i.e., flatten the grid into a vector and let each element be one feature), summary statistics (e.g., mean and standard deviation for each channel), and principal component loadings. The advantage of a CNN is that it learns simultaneously the best feature representation and the best mapping from features to predictions.

Weights in the dense layers are learned by gradient descent [Eq. (5)], simultaneously with those in the convolution layers. In this work, the activation function is softmax [section 21.5 in Russell and Norvig (2010)] for

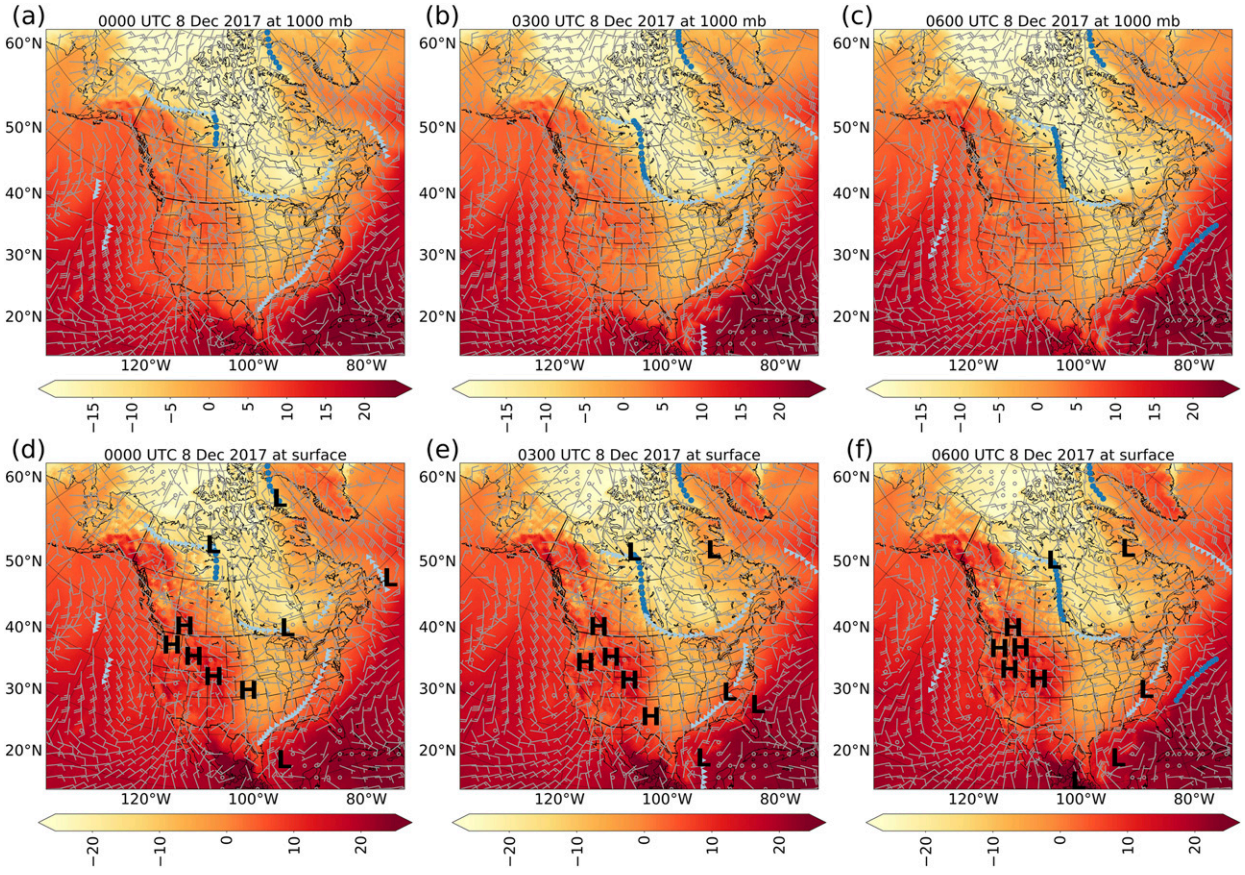

FIG. 5. WPC fronts from 0000 to 0600 UTC 8 Dec 2017. Gray vectors are wind barbs; the color fill is wet-bulb potential temperature (°C); light blue triangles are cold fronts; and dark blue circles are warm fronts. (a)–(c) The 1000-mb fields, determined to be the best vertical level for CNN-training (see discussion in section 5a). (d)–(f) Surface fields, on which the WPC bulletins are explicitly based. The letters "H" and "L" indicate analyzed low and high pressure centers, respectively, in the WPC bulletins. These examples show that the WPC dataset has temporal inconsistencies in both frontal existence (e.g., cold fronts in the Pacific) and morphology (e.g., warm front in central Canada).

the last dense layer and ReLU for the others. Softmax ensures that all three outputs are in the range [0, 1] and sum to 1.0, which allows them to be interpreted as probabilities. To our knowledge, no other activation function is appropriate for classification with more than two classes. Equation (6) compares softmax-generated probabilities to the true labels, yielding the loss for gradient descent.

3. Input data and preprocessing

Labels (or "ground truth") come from the Weather Prediction Center (WPC) surface bulletins (National Weather Service 2007). These bulletins are produced every three hours and contain a set of polylines, each demarcating a front (Fig. 5). WPC labels are generally on the synoptic scale (e.g., only 7% of warm fronts and 3% of cold fronts are shorter than 200 km), which fits the goal of this project. Also, when fronts are short it is often because several nearly collinear fronts are drawn through the same thermal transition zone (e.g., cold fronts in the eastern United States in Figs. 5b,c).

The WPC labels are created by human meteorologists, which introduces two types of inconsistency. The first is intrapersonal, where the same meteorologist applies different rules (definition of a front) to each case. The second is interpersonal, where different meteorologists have different rules. As a result, the WPC labels sometimes undergo dramatic morphological changes, or disappear and reappear, between successive time steps (e.g., Fig. 5). These issues notwithstanding, we use human labels rather than ones created by an algorithm such as NFA (section 1), because these algorithms have their own errors, which tend to be more simple and systematic. For example, Schemm et al. (2015) find that the thermal method rarely detects fronts with weak baroclinicity, while the wind-shift method rarely detects warm fronts. These biases would be easy for a CNN to mimic, which is tantamount to overfitting peculiarities of the training data. Since the WPC labels

are created by several humans, each with different tendencies, this dataset is much harder to overfit.

Another disadvantage of human labels is that they are expensive, because they require labor from people with rare expertise. Our labels already existed, but in general this is a problem for supervised ML (the type where correct answers are needed for training). When human labels are unavailable the best option may be labels from a set of algorithms with different biases (e.g., the thermal and wind-shift methods discussed in Schemm et al. 2015).

Predictors come from the North American Regional Reanalysis (NARR; Mesinger et al. 2006), which outputs data every three hours (synchronously with the WPC bulletins) on a 32-km grid. The 32-km spacing is adequate for merely detecting synoptic-scale fronts, but frontal zones are often narrower than 64 km (two grid lengths) (Roeder and Gall 1987), which hinders accurate placement. However, given the errors in the WPC dataset (Fig. 5), we do not think that higher-resolution data would provide any benefit. The NARR covers both the spatial extent (Fig. 6) and time span (Table 3) of the WPC bulletins. However, as shown in Figs. 6a,b, most WPC fronts are near the North American continent, which is a small subset of the NARR domain. Thus, we mask out grid cells with <100 WPC fronts, as shown in Fig. 6c. No masked grid cell is ever used as the center of a training example (one of the "$M \times N \times P$ images" described later in this section and shown at the top left of Fig. 2). The same applies for validation and testing.

The time period is split into training, validation, and testing. The role of training data is to fit the model (i.e., adjust the weights); the role of validation data is to compare models with different settings on unseen data and choose the best one; and the role of testing data is to provide an independent assessment of model performance on unseen data (used to neither fit nor validate the model). The main requirement of the three datasets is that they be statistically independent. For example, if the data were split randomly, an example from 0300 UTC today could fall into the training set, with 0600 UTC in the testing set. These data would probably be highly correlated (fronts usually do not evolve much in three hours), so the testing set would not provide an *independent* assessment of performance. With this in mind, we split the data as shown in Table 3, with a one-week gap between datasets.

We rotate the horizontal wind (u and v) from Earth relative to grid relative. Grid coordinates and wind coordinates should be the same, so that the relative orientation of wind vectors and thermal or height gradients are easier to infer. Perhaps the models could have learned equally well from Earth-relative winds, but this would be nontrivial, since the rotation angle is different at each grid cell.

To match the labels with predictor variables, we convert the WPC fronts from polylines to images on the NARR grid. The label at each grid cell—no front (NF), warm front (WF), or cold front (CF)—is based on the type of front intersecting it. This conversion resolves a problem inherent to polylines: they depict fronts as infinitesimally thin. However, there is still a representativity error due to the NARR's finite grid spacing. If a polyline intersects grid cell p near its edge with grid cell q, q will not be labeled as part of the front, even though it is probably in the frontal zone. To account for this error, we dilate fronts via the following procedure at each time step (Figs. 7a,b). Class frequencies before and after dilation are shown in Table 4.

1) Dilate each WF grid cell (i, j), using eight-connectivity. This means that all vertical, horizontal, and diagonal neighbors of (i, j) take the label WF. Eight-connectivity with 32-km spacing is equivalent to a ~50-km buffer.
2) Dilate all CF grid cells, using the same method.
3) For any grid cell labeled both WF and CF, replace with the nearest frontal label (WF *or* CF) in the undilated image. In case of a tie, the grid cell is labeled CF (the more common label in the undilated dataset). For example, this step is applied at both ends of the cold front from Lake Superior to western Québec in Figs. 7a and 7b.

Each training example consists of an $M \times N \times P$ image (top left of Fig. 2) and scalar target value. The variables M, N, and P are the number of rows, columns, and predictor variables in the image, respectively. The target value is the true label (NF, WF, or CF) after dilation at the center of the $M \times N$ grid. Terms M and N are always odd, so that there is a grid cell exactly at the center. Each predictor variable is normalized by Eq. (7). Here $x'_{i,j}$ and $x_{i,j}$ are the normalized and unnormalized values of predictor x at grid cell (i, j), respectively; and x_{p_1} and $x_{p_{99}}$ are the 1st and 99th percentiles of unnormalized x values in the NARR grid at the same time, respectively. We do not use the strict minimum and maximum, because this would allow outliers (which may be erroneous) to unduly influence the normalization. Equation (7) normalizes each variable to ~[0, 1]:

$$x'_{i,j} = \frac{x_{i,j} - x_{p_1}}{x_{p_{99}} - x_{p_1}}. \tag{7}$$

In a separate experiment (not shown) we normalized the images with a static x_{p_1} and $x_{p_{99}}$ (one for each predictor,

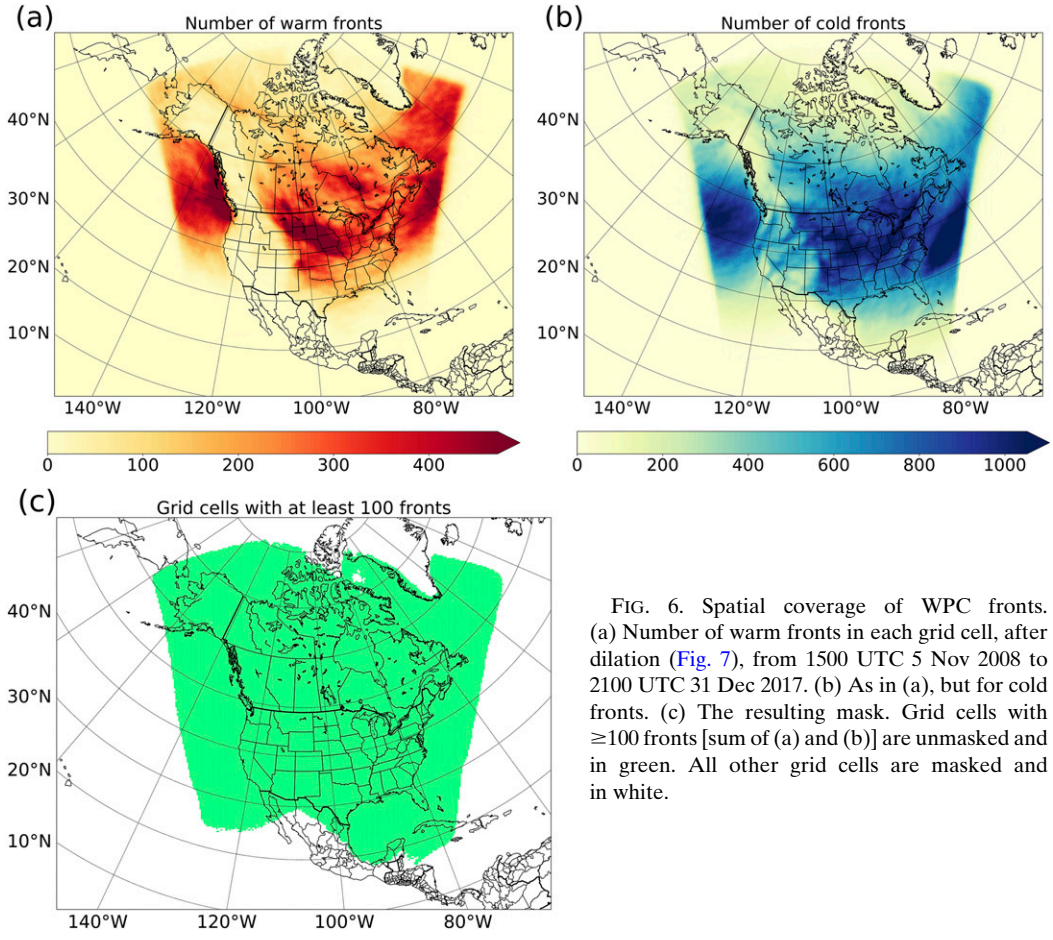

FIG. 6. Spatial coverage of WPC fronts. (a) Number of warm fronts in each grid cell, after dilation (Fig. 7), from 1500 UTC 5 Nov 2008 to 2100 UTC 31 Dec 2017. (b) As in (a), but for cold fronts. (c) The resulting mask. Grid cells with ≥100 fronts [sum of (a) and (b)] are unmasked and in green. All other grid cells are masked and in white.

rather than one for each predictor and time step). However, this worsened model performance. For thermal variables (T, q, and θ_w), using a different x_{p_1} and $x_{p_{99}}$ at each time step emphasizes thermal gradients in the summer, because there is less variation over the Northern Hemisphere, so the denominator in Eq. (7) is smaller. We hypothesize that this works better because it matches the way meteorologists think. During the summer, when thermal gradients are weaker, meteorologists are more ''generous'' in labeling fronts (the required thermal gradient is smaller).

4. Postprocessing and model evaluation

a. Gridcell-wise evaluation

In gridcell-wise evaluation the probability grid (Fig. 8a) is compared, grid cell by grid cell, with the target grid (containing true values). This type of evaluation has fallen out of favor in meteorology, because it unduly punishes slight offsets between the predictions and observations. For example, in Fig. 7c, the predicted front is shifted one grid cell to the north and east of the observed front, causing all grid cells in the predicted front to be counted as false positives and all those in the observed front to be counted as false negatives. This effect unduly punishes an otherwise-perfect prediction. However, for all training, validation, and testing, the target grids are dilated as in section 3. Probability grids need not be explicitly dilated, because the models are trained with dilated target grids, so dilation is automatically built into the predictions. Since dilation corresponds to a ~50-km

TABLE 3. Temporal data coverage (at the time we began experiments).

Dataset	Time period
NARR	0000 UTC 1 Jan 1979–2100 UTC 31 Dec 2017
WPC fronts	1500 UTC 5 Nov 2008–1200 UTC 18 Jan 2018
Training period	1500 UTC 5 Nov 2008–2100 UTC 24 Dec 2014
Validation period	0000 UTC 1 Jan 2015–2100 UTC 24 Dec 2016
Testing period	0000 UTC 1 Jan 2017–2100 UTC 31 Dec 2017

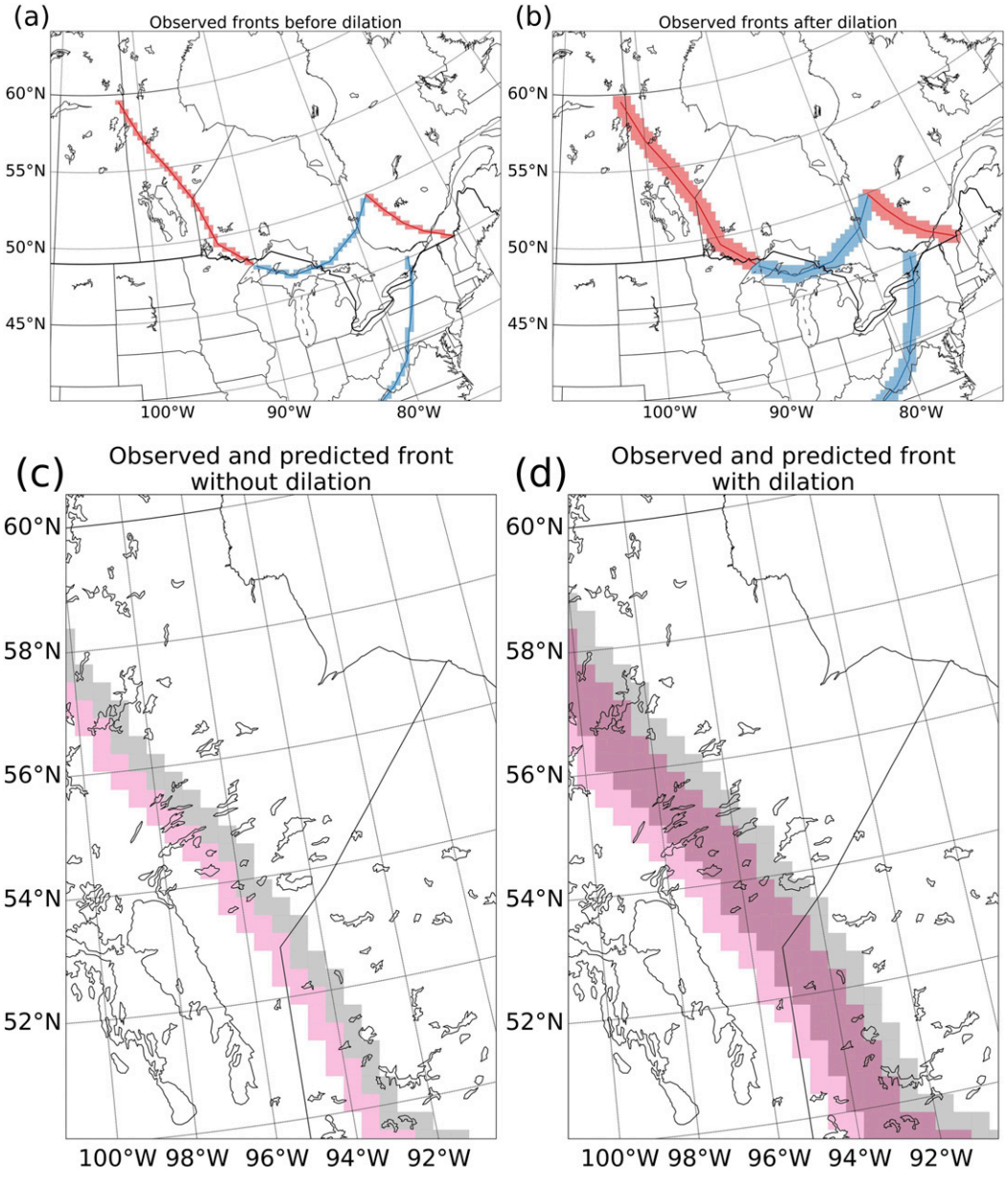

FIG. 7. Dilation and its role in gridcell-wise evaluation. (a) Undilated and (b) dilated WPC fronts at 0000 UTC 1 Dec 2017. Warm fronts are in red, and cold fronts are in blue. Dark lines are WPC fronts, and light shading shows NARR grid cells intersected by the fronts before and after dilation. (c) Observed and predicted front without dilation; (d) with dilation. Grid cells intersected by only the predicted front (gray) are counted as false positives; those intersected by only the observed front (light pink) are counted as false negatives; those intersected by both (dark pink) are counted as true positives.

buffer, gridcell-wise evaluation with dilation is similar to neighborhood evaluation with a 50-km neighborhood distance, which does not unduly punish slight offsets (Fig. 7d).

The performance metrics used for this study (appendix) are based on the contingency table (Table 5), which requires deterministic predictions. However, our CNNs (like most ML models for classification) output probabilities. Thus, we determinize the probabilistic predictions via the following procedure, illustrated in Figs. 8a,b.

1) Round all NF (no front) probabilities in the validation set to the nearest 0.001 and create an array of unique values. This eliminates very similar values and reduces the amount of computation required in step 2.

TABLE 4. Class frequencies before and after dilation (section 3) for the study period (1500 UTC 5 Nov 2008–2100 UTC 31 Dec 2017). Each value is the average number of NARR grid cells at one time intersected by no front (NF), a warm front (WF), or a cold front (CF).

Class	Before dilation	After dilation
NF	99.66%	98.95%
WF	0.09%	0.27%
CF	0.25%	0.78%

2) For each value p^*_{NF} in the unique array, apply Eq. (8). Here p_{NF}, p_{WF}, and p_{CF} are predicted probabilities of the three classes; P is the resulting deterministic prediction:

$$P = \begin{cases} \text{NF}, & p_{NF} \geq p^*_{NF}; \quad \text{otherwise:} \\ \text{CF}, & p_{CF} \geq p_{WF} \\ \text{WF}, & p_{CF} < p_{WF} \end{cases} . \qquad (8)$$

The threshold p^*_{NF} is applied to the NF class because, due to the imbalance in class frequencies (Table 4), models generally produce higher p_{NF} than p_{WF} or p_{CF}. For example, if the threshold were applied to p_{WF}, the second line of Eq. (8) would be "NF, $p_{NF} \geq p_{CF}$", leading to very few deterministic CF predictions. If the threshold were applied to p_{CF}, the second line would be "NF, $p_{NF} \geq p_{WF}$", leading to very few deterministic WF predictions. Thus, the best strategy is to determine if there is *any* front [first line of Eq. (8)], then determine its type by comparing p_{WF} and p_{CF}.

3) Find the optimal p^*_{NF}, which is the one that produces the highest Gerrity score [Eq. (A4), used for reasons discussed later in this section]. The optimal threshold is based on validation data, and the same threshold is used later on testing data.

Deterministic predictions from step 3 are used to create contingency tables, which are used to compute accuracy [Eq. (A1)], the Heidke score [Eq. (A2)], Peirce score [Eq. (A3)], and Gerrity score [Eq. (A4)]. Accuracy varies from [0, 1]; the Heidke score varies from $(-\infty, 1]$; and the Peirce and Gerrity scores vary from $[-1, 1]$. Higher is better in all cases. For the Heidke, Peirce, and Gerrity scores, 0.0 indicates no skill. The loss function used during training [Eq. (6)] and performance metrics used after training are different, because the former evaluate probabilities and the latter evaluate deterministic predictions. The performance metrics would make poor loss functions, because computing p^*_{NF} after each weight update would be expensive and determinizing the predictions would make derivatives in Eq. (5) hard to define.

The main disadvantage of accuracy is that for rare events it can be "played" by always predicting the majority class. For this study, a model that always predicts

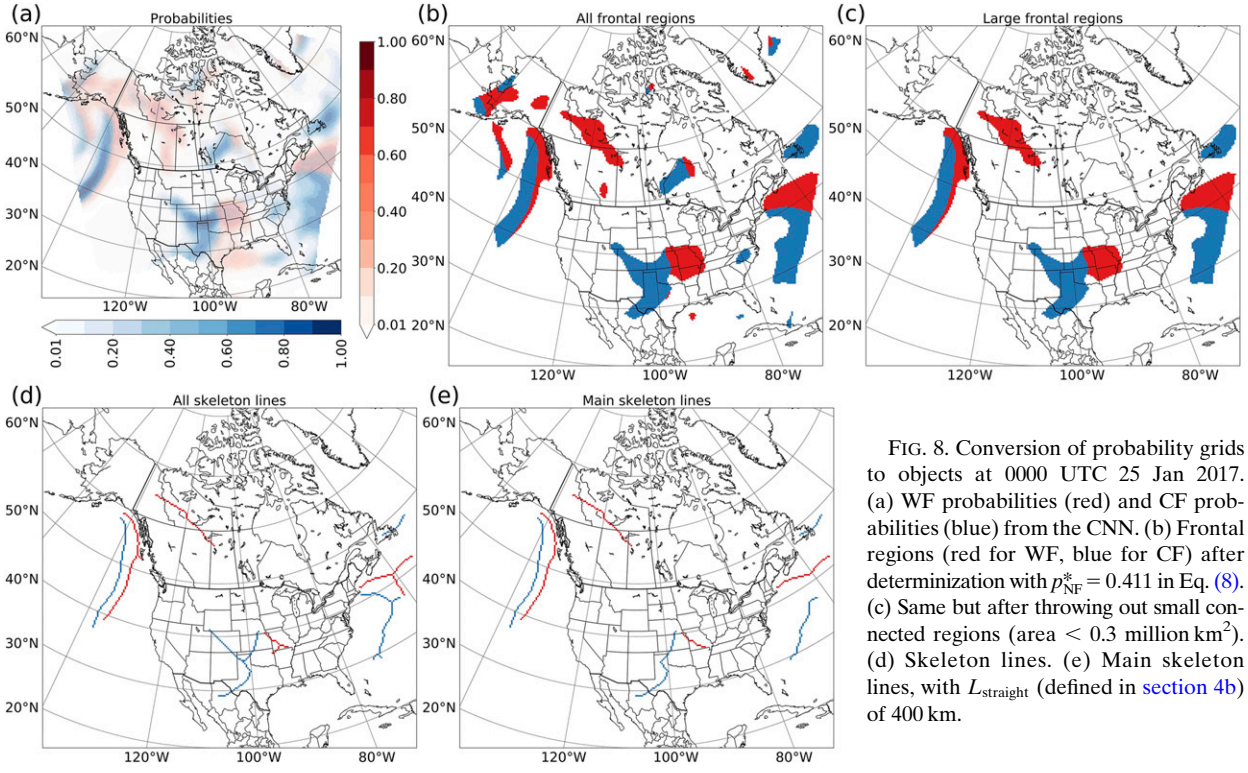

FIG. 8. Conversion of probability grids to objects at 0000 UTC 25 Jan 2017. (a) WF probabilities (red) and CF probabilities (blue) from the CNN. (b) Frontal regions (red for WF, blue for CF) after determinization with $p^*_{NF} = 0.411$ in Eq. (8). (c) Same but after throwing out small connected regions (area < 0.3 million km^2). (d) Skeleton lines. (e) Main skeleton lines, with $L_{straight}$ (defined in section 4b) of 400 km.

TABLE 5. Contingency table for three-class prediction; n_{ij} is the number of examples where the ith class is predicted and the jth class is observed. In a perfect contingency table, $n_{ij} = 0$ if $i \neq j$.

	Observed		
Predicted	NF	WF	CF
NF	n_{11}	n_{12}	n_{13}
WF	n_{21}	n_{22}	n_{23}
CF	n_{31}	n_{32}	n_{33}

NF would have 98.95% accuracy, as shown in Table 4. The Heidke and Peirce scores measure the fraction of correct predictions, excluding those that would arise from random and climatological guessing, respectively. Their main disadvantage is that the random and climatological baselines are naïve and easily outperformed. Finally, the Gerrity score is equitable (gives random and constant predictions a score of 0.0) and rewards correct predictions of the minority classes (WF and CF) more than the majority class (NF). This second property means that, unlike the Heidke and Peirce scores, the Gerrity score does not reward conservative prediction (''erring on the side of the majority class''). This is why we use the Gerrity score in step 3 of the determinization procedure. This causes the CNNs to overpredict fronts, as manifested in the large frontal regions shown in Fig. 8b. However, this overprediction is mitigated by object conversion (section 4b), especially the skeletonization step (Fig. 8d).

b. Converting probability grids to objects

The probability grids have two disadvantages. First, the apparent frontal zones are unusually wide (Fig. 8b). This can be mitigated by decreasing the determinization threshold [p^*_{NF} in Eq. (8)], but at the cost of creating many small and nonconnected frontal zones, which is also undesirable. Second, humans generally think of fronts as objects rather than grids, so a useful front-detection algorithm would produce explicit objects, like most NFA methods discussed in section 1. We use the following procedure to achieve this.

1) Convert the images to connected regions. A connected region is a set of eight-connected WF or CF grid cells, with ''eight-connectivity'' defined as in section 3. Although this does not change the appearance of the map (Fig. 8b), it changes the internal representation from a grid to a collection of objects.
2) Throw out small frontal regions (with area below a threshold; Fig. 8c). Small regions are less likely to represent synoptic-scale fronts and more likely to be false alarms.

TABLE 6. Parameters for Experiment 1 (CNN-training). The pressure level for all predictor variables is 1000 mb, and the ''image size'' is the spatial dimensions of each input example.

Parameter	Values
Predictor variables	u, v, and θ_w
	u, v, T, and q
	u, v, θ_w, T, and q
	u, v, θ_w, and Z
	u, v, T, q, and Z
	u, v, θ_w, T, q, and Z
Image size	9×9, 17×17, 25×25, 33×33
Dropout fraction	0.25, 0.5

3) Reduce each region to a one-gridcell-wide skeleton (Fig. 8d). This is done by morphological thinning,[2] a common image-processing algorithm.
4) The resulting skeletons are usually complex polylines (with more than two endpoints) and often have much more complicated shapes than human-analyzed fronts. Thus, we split each skeleton into simple skeletons and find the main skeleton. A ''simple skeleton'' is a simple polyline contained entirely within, and connecting two endpoints of, the original skeleton. For a skeleton with K endpoints, there are $(1/2)K(K-1)$ simple skeletons. The ''main skeleton'' is that with the greatest adjusted length, defined in Eq. (9). Here, L_{straight} is the distance between the two endpoints; L_{int} is the integrated length of the polyline (the sum of all line segments); and $S = L_{\text{int}}/L_{\text{straight}}$ is the sinuosity. Dividing by sinuosity discourages complicated shapes, such as ''V'' shapes, which occur frequently and do not resemble human-analyzed fronts. Before computing L_{adj} for each simple skeleton, we throw out those with $L_{\text{straight}} < L^*_{\text{straight}}$, where L^*_{straight} is a user-selected parameter:

$$L_{\text{adj}} = \frac{L_{\text{straight}}}{S} = \frac{L^2_{\text{straight}}}{L_{\text{int}}}. \tag{9}$$

Objects produced by step 4 (Fig. 8e) are considered ''predicted fronts.''

c. Object-based evaluation

Object-based evaluation compares actual (WPC) fronts with predicted fronts, using a neighborhood distance.[3] Warm fronts can be matched only to other

[2] https://scikit-image.org/docs/dev/api/skimage.morphology.html#skimage.morphology.thin.

[3] The distance between two fronts is defined as the median distance, over all points P in one front, between P and the nearest point in the other front.

TABLE 7. Parameters for Experiment 2 (object conversion). Determinization thresholds are at intervals of 0.05 around the best threshold for gridcell-wise prediction (0.411), found in Experiment 1.

Parameter	Values
Determinization threshold [p^*_{NF} in Eq. (8)]	0.261, 0.311, 0.361, 0.411, 0.461, 0.511, 0.561, 0.611, 0.661, 0.711, 0.761, 0.811
Minimum region area ($\times 10^6$ km^2)	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
Minimum front length (km) ($L^*_{straight}$ in section 4b)	100, 200, 300, 400, 500, 600, 700, 800, 900, 1000

warm fronts, and likewise for cold fronts. A key difference between object-based and gridcell-wise evaluation is that in the latter, because each example is one grid cell and all grid cells are classified in both the actual and predicted datasets, ''negative examples'' (NF grid cells) are well defined. In object-based evaluation there are only WF and CF objects, no NF objects, so ''negative examples'' are undefined. Most three-class performance metrics, including those used in section 4a, break down in this setting. However, some two-class performance metrics can still be computed; for example, Eqs. (A5)–(A8) are used by the National Weather Service to verify tornado warnings (page 1, Brooks 2004), another setting in which negative examples (''nontornadoes'') are difficult to define.

5. Experimental setup

a. Experiment 1: CNN experiment

This experiment determines the best input data and CNN architecture for gridded classification. Specifically, we try all 48 combinations of the parameters listed in Table 6. The predictor variables always include u and v, because otherwise the model identifies many stationary fronts, which are generally not labeled in the WPC data and therefore count as false alarms. Predictors may also include the fundamental thermal variables (T and q), θ_w (which combines information from T and q), and/or Z (because fronts are often collocated or nearly collocated with a height trough). When predicting the label for a training example, $d \times 100\%$ of dense-layer weights are omitted, where d is the dropout fraction. This forces dense-layer weights to adapt more independently of each other, which reduces overfitting. Dropout is not used for validation or testing.

The number of filters in the first convolution block (Fig. 2) is always 8 $\times$ the number of predictor variables (we have found subjectively that this works well for many problems). The number of filters doubles with each successive convolution block, which is a common practice. The number of convolution blocks is two for images smaller than 25 $\times$ 25 (three otherwise). For smaller images, three blocks would cause the last feature maps to be 1 $\times$ 1 or 2 $\times$ 2, thus not containing enough information to make skillful predictions. Image sizes themselves are chosen to determine how much local context is needed to label the center grid cell. We cap image size at 33 $\times$ 33 (1056 km $\times$ 1056 km), because data over 500 km away should not be needed to make this decision. Also, in preliminary work larger images often caused the CNN to collapse, as it does for 33 $\times$ 33 images with six predictors (see discussion in section 6a).

Each model is trained for 100 epochs, with 32 batches per epoch, each containing 1024 examples (e.g., the top-left panel of Fig. 2 is one example). Training data are downsampled: each batch contains 512 NF, 256 WF, and 256 CF examples. Because 98.95% of examples are NF (Table 4), without downsampling the model has little incentive to predict any label other than NF. Downsampling is used only for training, not for validation or testing. Thus, in validation and testing the class frequencies are approximately as listed in Table 4.

The weight update [Eq. (5)] is performed for one batch at a time. The 1024 examples are randomly drawn from 128 times in the training period. Including many examples and many times makes the batch diverse, which prevents overfitting. As a counterexample, suppose that each batch contained examples from only one time. At 0000 UTC (Fig. 5a) the model would have to learn that there are no fronts in the Pacific; at 0300 UTC (Fig. 5b) it would have to learn that there are two cold fronts here; and at 0600 UTC (Fig. 5c) it would have to relearn that there are no fronts. This would make training unstable (weights would oscillate rather than converge). We chose 100 epochs and 32 batches per epoch because we found that this is more than enough time for convergence.

All predictors are taken from the 1000-mb pressure level. In an earlier experiment (not shown) we trained CNNs with similar architectures on fields at 900 mb, 950 mb, 1000 mb, and the surface.[4] The 1000-mb model achieved the best validation Gerrity scores. Finally, all

[4] Heights were 2 m above ground level (m AGL) for temperature, specific humidity, and θ_w; 10 m AGL for wind; and 2 m AGL for pressure. We replaced geopotential height with pressure in these CNNs.

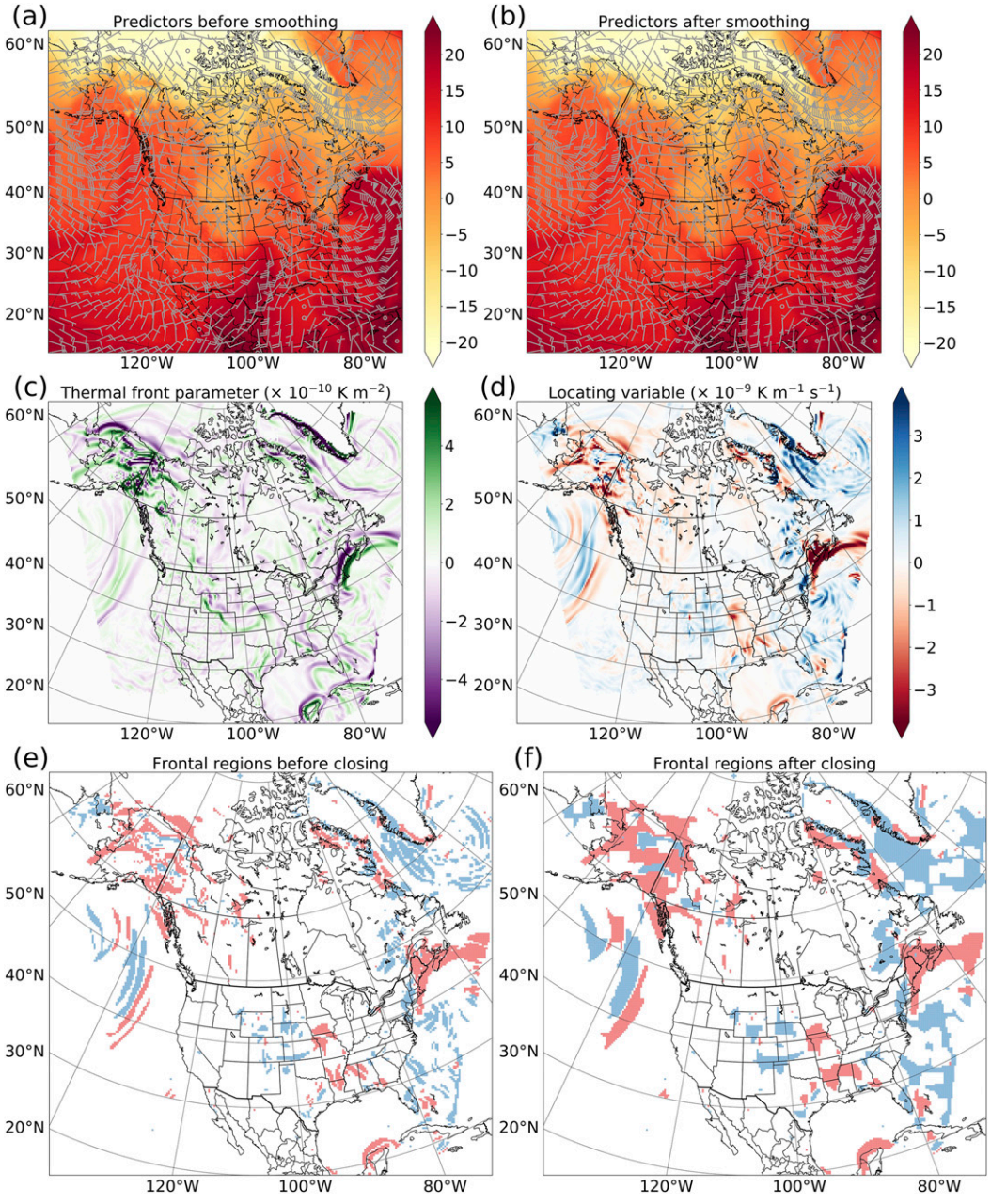

FIG. 9. Numerical frontal analysis at 0000 UTC 25 Jan 2017. (a) Original NARR fields at 900 mb (formatted as in Fig. 5). (b) NARR fields after Gaussian smoothing with 32-km radius. (c) TFP [Eq. (1)]. (d) Locating variable [Eq. (10)]. (e) Front labels (red for warm front, blue for cold front, FP = 96 in section 5c). (f) As in (e), but after two iterations of binary closing.

models use L_2 regularization for the convolutional layers, with a strength of 0.001. This adds $0.001w$ to the loss function for each weight w [element of $\mathbf{W}_i^{(j,k)}$ in Eq. (4)], leading to fewer large weights, a simpler model, and less overfitting. We have found subjectively that 0.001 works well for many problems.

This experiment has many fewer trials (48) than Experiments 2 and 3 (1200 and 3600, respectively), because training CNNs is more computationally expensive. Training one CNN takes ~48 h on seven CPU cores.

b. *Experiment 2: Object-conversion experiment*

This experiment finds the best object-conversion parameters for the best CNN from Experiment 1. Specifically, we try all 1200 combinations of the

parameters listed in Table 7, which are explained in section 4b. Object conversion is completely determined by the parameters in Table 7 and, unlike a CNN, does not involve learned weights. Thus, the object-conversion experiment requires only validation and testing data, no training data. We find the best parameters, that is, those yielding the highest critical success index [CSI; Eq. (A8)] on validation data (Table 3). This experiment uses only the best CNN from Experiment 1, rather than all CNNs, which would be too computationally expensive. Creating a full prediction grid, with one classification for each of the 41 591 unmasked NARR grid cells, takes 5–10 min per CNN.

We use CSI to select the best parameters because, as noted in section 4c, correct nulls do not exist in the object-based setting, which limits the number of performance metrics that can be computed. Among those that can still be computed are POD, success ratio, frequency bias, and CSI [Eqs. (A5)–(A8)]. POD can be trivially optimized by predicting a front everywhere; success ratio can be trivially optimized by never predicting a front; and frequency bias can be perfect (1.0) even if both the POD and success ratio are very low. CSI is the only one of these metrics that cannot be trivially optimized, because perfect CSI requires perfect POD and success ratio (as shown in Fig. 13).

c. Experiment 3: Baseline experiment

The purpose of this experiment is to provide a non-ML baseline against which to compare our model. This is motivated by the belief that using ML rather than expert systems (such as NFA), which are generally easier to understand, should be justified by superior performance. To our knowledge, none of the previous work in NFA (section 1) includes objective evaluation on a large number of examples; most include subjective evaluation on only a few case studies. Thus, we develop our own NFA method, which is similar to those developed in previous work. To create a prediction grid (analogous to Fig. 8b), we use the following procedure.

1) Apply a Gaussian smoother to each of the three NARR fields[5] (u, v, and θ_w). Without smoothing, the derivatives computed in step 2 are very noisy. See Figs. 9a and 9b.
2) Compute the TFP [Eq. (1) with $\tau = \theta_w$] at each unmasked grid cell (Fig. 6c). See Fig. 9c.
3) Compute the locating variable [Eq. (10)] at each unmasked grid cell. Here, |TFP| is the absolute value of TFP ($\mathrm{K\,m^{-2}}$); $\mathbf{v} = (u, v)$ is the horizontal wind vector ($\mathrm{m\,s^{-1}}$); $\mathbf{v} \cdot \hat{\nabla}\theta_w$ is the horizontal wind speed in the direction of the thermal gradient, where positive values indicate cold-air advection and negative values indicate warm-air advection; and LV is the locating variable ($\mathrm{K\,m^{-1}\,s^{-1}}$; see Fig. 9d):

$$\mathrm{LV} = |\mathrm{TFP}|\mathbf{v} \cdot \hat{\nabla}\theta_w . \tag{10}$$

4) Determine the front type at each grid cell. This is based on two thresholds: $\mathrm{LV}^*_{\mathrm{cold}}$, which is percentile FP of all positive values in the grid, and $\mathrm{LV}^*_{\mathrm{warm}}$, which is percentile (1 − FP) of all negative LV values in the grid. FP is an input parameter, and the label (P) for each grid cell is determined by Eq. (11):

$$P = \begin{cases} \mathrm{CF}, & \mathrm{LV} \geq \mathrm{LV}^*_{\mathrm{cold}} \\ \mathrm{WF}, & \mathrm{LV} \leq \mathrm{LV}^*_{\mathrm{warm}} \\ \mathrm{NF}, & \text{otherwise} \end{cases} . \tag{11}$$

   The thresholds are computed independently for each grid (time step), for the same reason that normalization is done independently at each time step [see the explanation following Eq. (7)]. See Fig. 9e.
5) Use binary closing[6] to connect nearby WF and CF regions. WF regions are connected only to other WF regions, and CF regions are connected only to other CF regions. This fills small gaps where LV does not quite meet the threshold. See Fig. 9f.

Finally, to convert the prediction grids to objects, we use the same procedure as in section 4b and Fig. 8. This experiment uses all 3600 combinations of the parameters listed in Table 8. We use smaller area thresholds than in Experiment 2, because the baseline method generally produces narrower frontal zones (cf. Figs. 8b and 9f).

TABLE 8. Parameters for Experiment 3 (baseline). Smoothing radius is the standard deviation for the Gaussian kernel; the front percentile is FP in section 5c; and minimum front length is L^*_{straight} in section 4b. When the "pressure level" is the surface, NARR variables used are 10-m wind and 2-m θ_w.

Parameter	Values
Smoothing radius (km)	32, 64
Front percentile	96, 97, 98, 99
Number of binary-closing iterations	1, 2, 3
Pressure level (mb)	900, 950, 1000, surface
Minimum region area ($\times 10^3\,\mathrm{km^2}$)	20, 40, 60, 80, 100
Minimum front length (km)	100, 200, 300, 400, 500, 600, 700, 800, 900, 1000

[5] In a separate experiment we tried ensembling NFA grids computed for three different thermal variables: T, q, and θ_w. However, the final validation CSI (0.2285) was the same in the experiment shown, to the fourth decimal point.

[6] https://scikit-image.org/docs/dev/api/skimage.morphology.html#skimage.morphology.binary_closing.

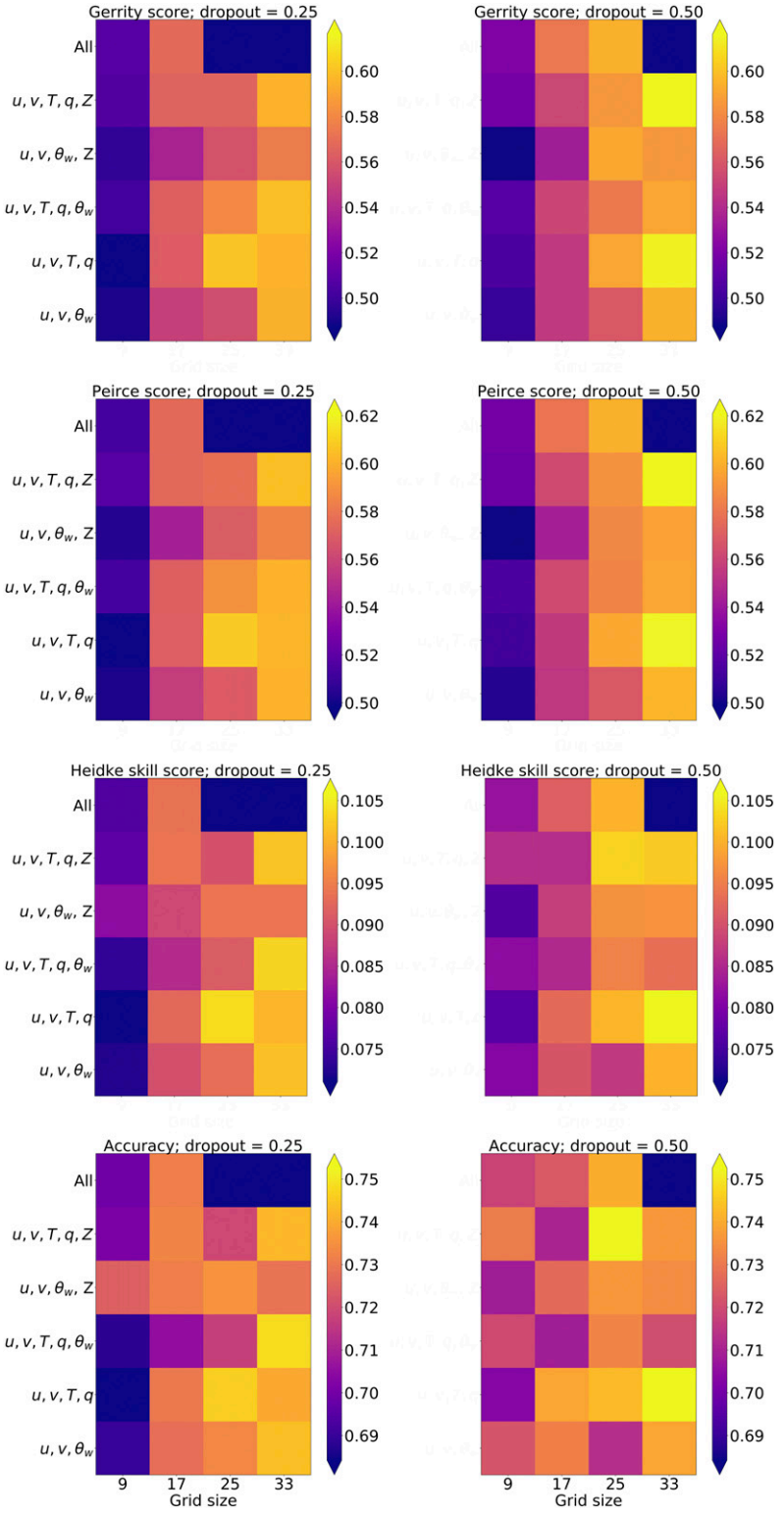

FIG. 10. Validation results for Experiment 1. Each panel corresponds to one performance metric and one dropout fraction, indicated in the title. Labels on the *y* axis indicate predictor variables; "all" means u, v, T, q, θ_w, and Z.

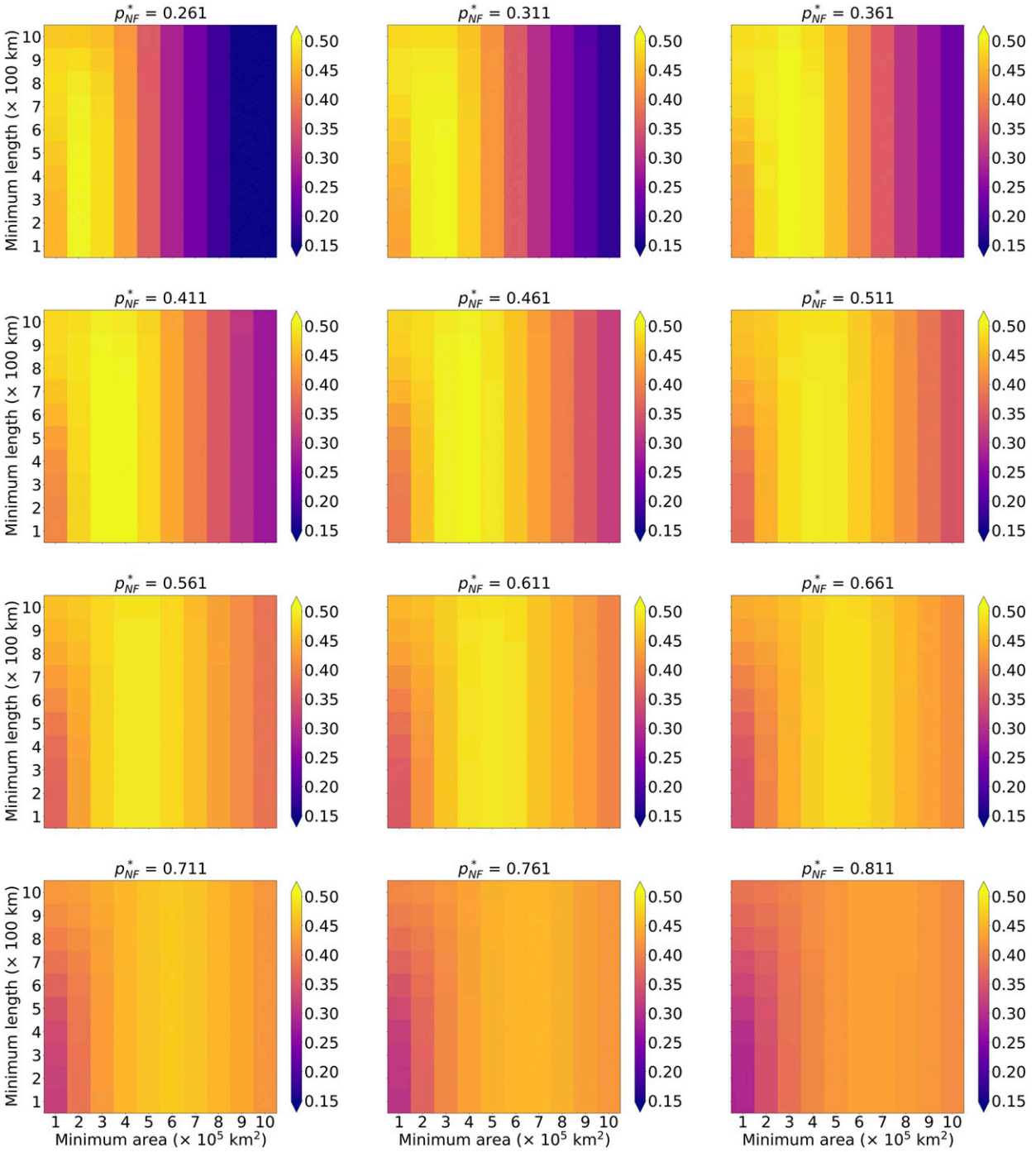

FIG. 11. Validation CSI for Experiment 2. Each panel corresponds to one determinization threshold [p^*_{NF} in Eq. (8)], indicated in the title.

6. Experimental results

a. Results on validation data

Validation results for Experiment 1 are based on 1 million examples drawn randomly from the validation period (Table 3). The same 1 million examples are used for each CNN, to ensure a fair comparison. Similarly, validation results for Experiment 2–3 are based on the same 1000 time steps drawn randomly from the validation period. Also, results for Experiments 2–3 are based on a 250-km neighborhood distance (section 4c). We also ran the experiments with a

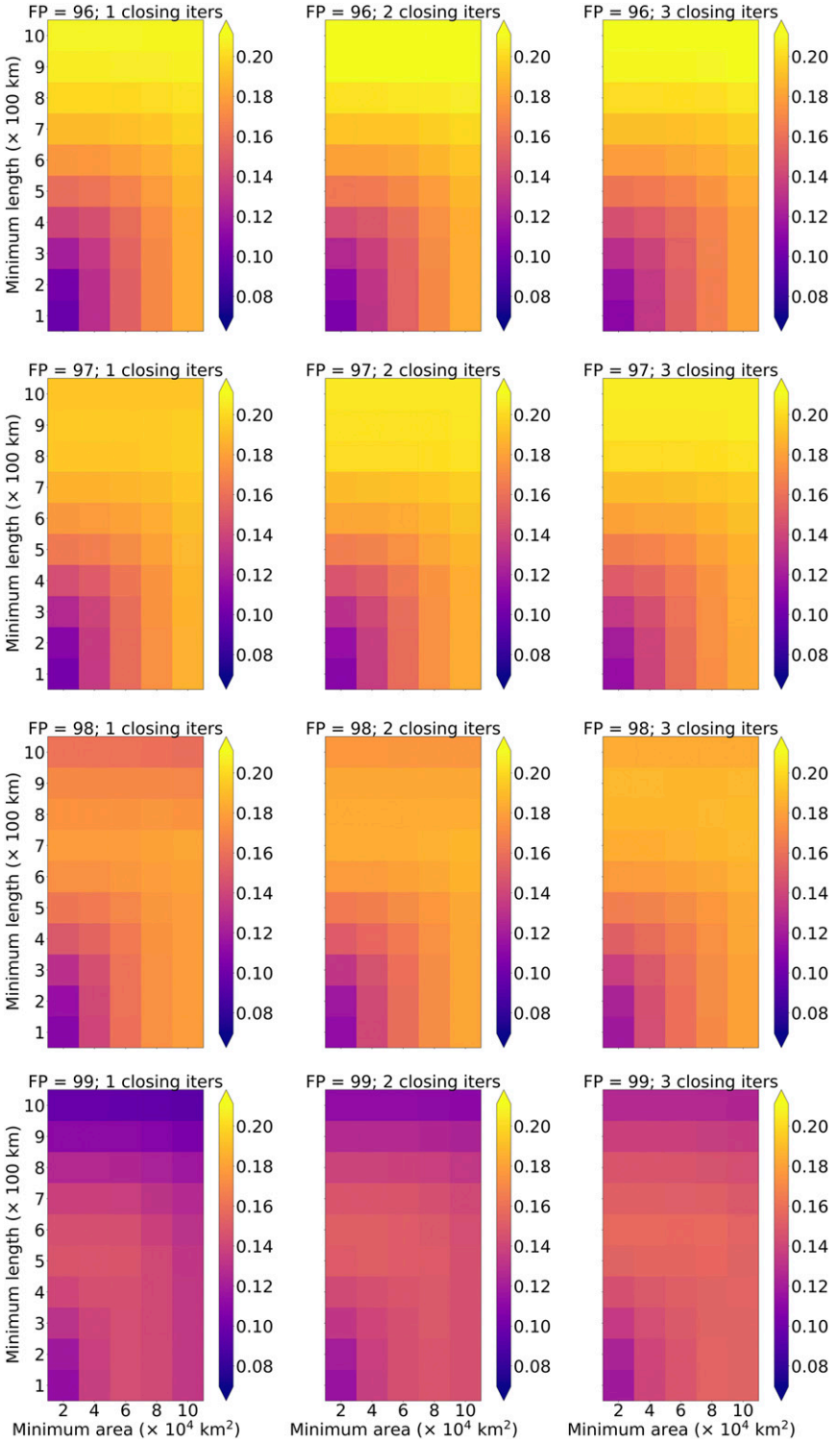

FIG. 12. Validation CSI for Experiment 3 with 32-km smoothing radius and 900-mb pressure level. Each panel corresponds to one front percentile (FP in section 5c) and one number of binary-closing iterations, indicated in the title.

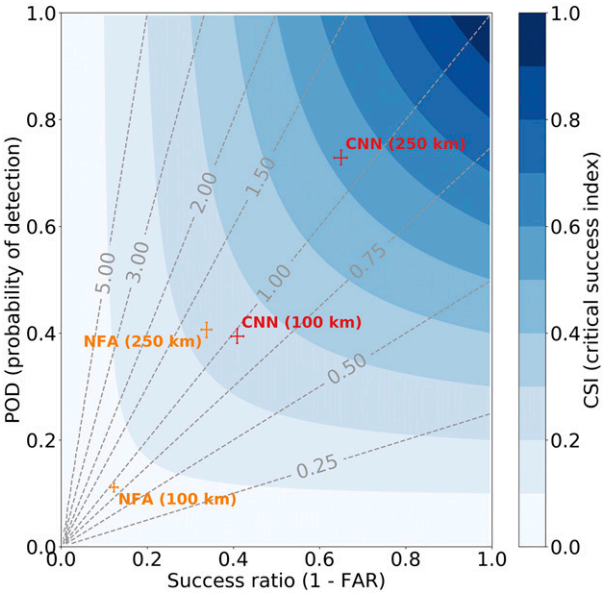

FIG. 13. Performance diagram (Roebber 2009) for the best CNN from Experiment 2, and best NFA model from Experiment 3, on testing data. Results are shown for both 100- and 250-km neighborhood distances. Dashed gray lines show frequency bias. All quantities shown (POD, success ratio, frequency bias, and CSI) are defined as in Eqs. (A5)–(A8). Error bars show the 99% confidence interval, determined by bootstrapping (Efron 1979) the testing set 1000 times.

100-km neighborhood distance (Fig. 13), resulting in much worse performance.

Results shown in Fig. 10 are based on deterministic predictions, using the determinization threshold [p^*_{NF} in Eq. (8)] that yields the highest validation Gerrity score. All four metrics—accuracy, Heidke, Peirce, and Gerrity scores—tend to increase with grid size. This suggests that larger grids—33×33 and 25×25, which are 800 km $\times$ 800 km and 1056 km $\times$ 1056 km, respectively—contain more useful information than the smaller ones. This is perhaps because fronts have varying strength (e.g., thermal-gradient magnitude or thermal advection) along their length. If a small grid contains only the weakest part of a front, the CNN may fail to identify the front. However, if the grid is expanded without changing its center point, the CNN might use information near the edges to identify that a front passes through the center grid cell, even if the front is weaker at that location.

The four metrics also improve as dropout fraction increases from 0.25 to 0.5, which suggests that overfitting occurs easily for our dataset and needs to be strongly mitigated. Finally, there is no discernible overall trend in performance with respect to the predictor variables. However, it is interesting that the two best models (those with the highest Gerrity scores) do not include θ_w as predictors. The other four models with the same grid size (33×33) and dropout fraction (0.5), all of which include θ_w, perform worse than the top two. This suggests that, although θ_w has been used in many studies of numerical and manual frontal analysis (section 1), it is not prioritized by WPC meteorologists.

The number of weights per CNN varies from 127 835 (for 9×9 images with 3 predictors) to 2 187 203 (for 33×33 images with all 6 predictors). After downsampling (section 5a), there are 17 880 115 training examples. For the largest CNN the ratio of examples to weights (~8) may be too small, causing their poor performance (top-right grid cell in each panel of Fig. 10).

For Experiment 2 (Fig. 11), CSI generally increases with lower p^*_{NF}, the determinization threshold in Eq. (8). Lower p^*_{NF} is more restrictive (results in more NF grid cells, so fewer WF and CF grid cells). CSI also decreases with minimum front length ($L^*_{straight}$ in section 4b), suggesting that a less restrictive value is better. Finally, for p^*_{NF} = 0.411 (the best determinization threshold for both Experiments 1 and 2), the best CSI occurs for a minimum region area of 0.3–0.4 million km^2. However, the location of this maximum changes with p^*_{NF}, increasing to 0.7 million km^2 for p^*_{NF} = 0.811. This suggests that higher (less restrictive) p^*_{NF} values need to be offset with higher (more restrictive) minimum areas, which makes sense because high p^*_{NF} values result in very large frontal regions.

The region of the parameter space with highest CSI (yellow pixels in Fig. 11) also tends to have the best frequency bias (near 1.0), which means that the number of predicted fronts is close to the number of actual fronts. The frequency bias before object conversion (for the best CNN from Experiment 1) is >10, because the CNN makes frontal regions too wide (Fig. 8b), but this is mitigated by object conversion, mainly skeletonization (Fig. 8d).

For Experiment 3, the best smoothing radius and pressure level are 32 km and 900 mb, respectively. This suggests that NFA performs better with data farther aloft, which require little smoothing, than with more heavily smoothed data near the surface. As mentioned in section 5a, an earlier experiment obtained the opposite result for CNNs: CNNs trained with 1000-mb data outperformed those trained with 900 and 950 mb. This suggests that CNNs handle noisy data better than NFA, which is not surprising. A known property of deep learning is its robustness to noisy data (Krause et al. 2016), while a known property of expert systems is their lack thereof (Ravuri et al. 2018).

Figure 12 shows results only for the best smoothing radius and pressure level, since showing all radii and pressure levels would require 84 more panels. CSI increases with minimum region area and minimum front length, so more restrictive values of these parameters

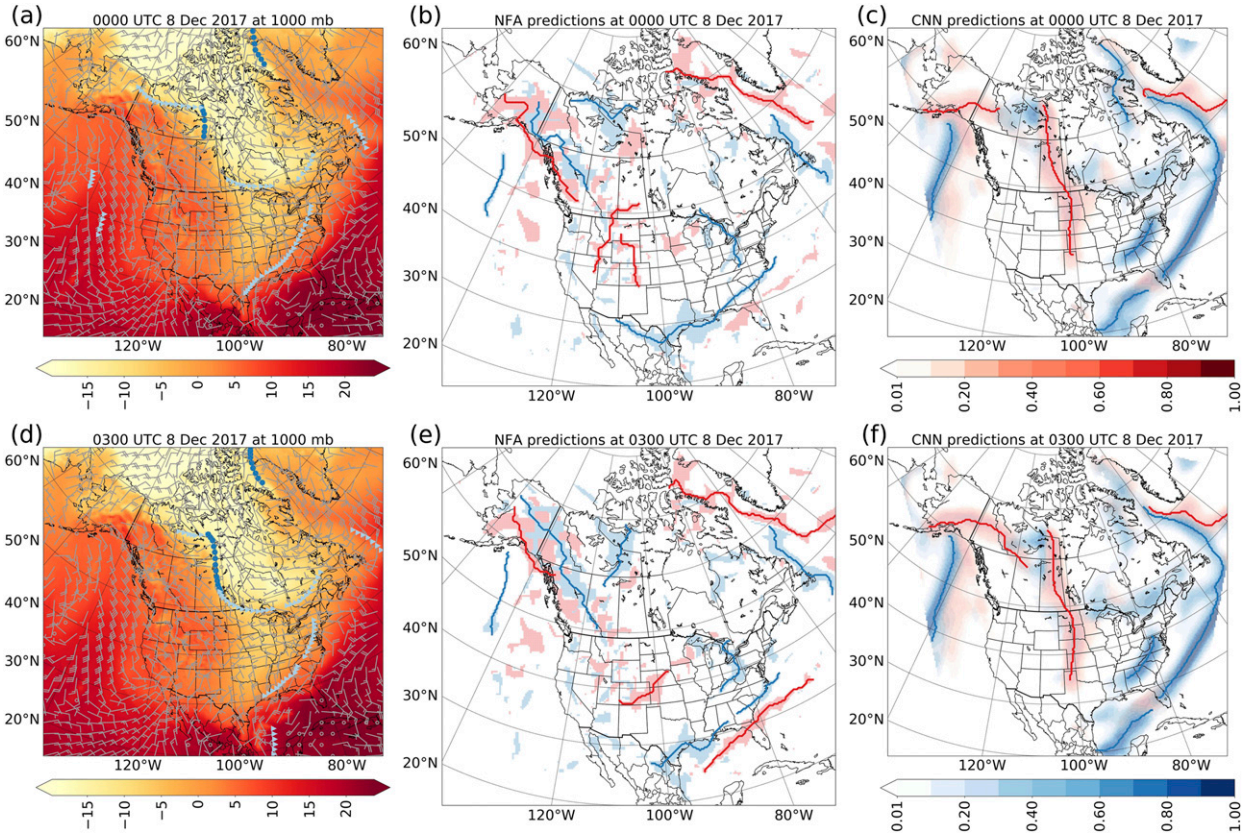

FIG. 14. Predictions of the best CNN (from Experiments 1 and 2) and best NFA method (from Experiment 3) for two time steps. (a) Predictors at 0000 UTC, formatted as in Fig. 5. (b) NFA predictions for 0000 UTC. WF predictions are in red; CF predictions are in blue; gridded predictions are in the color fill; and predicted objects are shown with thick lines. Gridded predictions are deterministic. (c) CNN predictions for 0000 UTC, formatted as in (b), except that gridded predictions are probabilistic (see color bars). (d)–(f) As in (a)–(c), except for 0300 UTC.

are preferred. This makes sense, given that the NFA method used to label each grid cell (section 5c) generally results in noisier fields than the CNNs (cf. Figs. 8 and 9), so only very large regions are likely to be actual fronts. Also, CSI increases with decreasing (less restrictive) values of FP (front percentile in section 5c). Thus, the baseline method generally performs better when the NFA parameters (used to label individual grid cells) are less restrictive, resulting in overprediction that is mitigated by object conversion with more restrictive parameters. Finally, CSI is maximized with two iterations of binary closing (section 5c), which suggests that two is just enough to join nearby regions that are part of the same front. By manual inspection, we found that three iterations often join regions corresponding to completely different fronts, which reinforces that two is a happy medium.

b. *Results on testing data*

Testing results are based on 1000 time steps, randomly drawn once from the testing period (Table 3). Figure 13 shows testing results for the best methods (those with the highest validation CSI) from Experiments 2 and 3. The CNN outperforms NFA, at the 99% confidence level, in all performance metrics except frequency bias.

Figures 14–15 compare the WPC fronts, best CNN with object conversion (from Experiment 2), and best NFA method (from Experiment 3) for four random times in the testing period. The times in Fig. 14 are from Fig. 5, where the WPC fronts have a lot of fine-scale detail and are inconsistent over time. Figure 15 shows more typical cases, where the WPC fronts contain less finescale detail (except that in the western United States) and are more consistent over time. Subjectively, gridded probabilities from the CNN match the WPC fronts better than gridded predictions from NFA or either set of predicted objects. Also, the CNN generates many fewer small regions than NFA (e.g., from 20° to 40°N in the Pacific). This is probably because NFA is based on local gradients, while the CNN is based on convolution at different resolutions (Fig. 2), so the CNN considers a

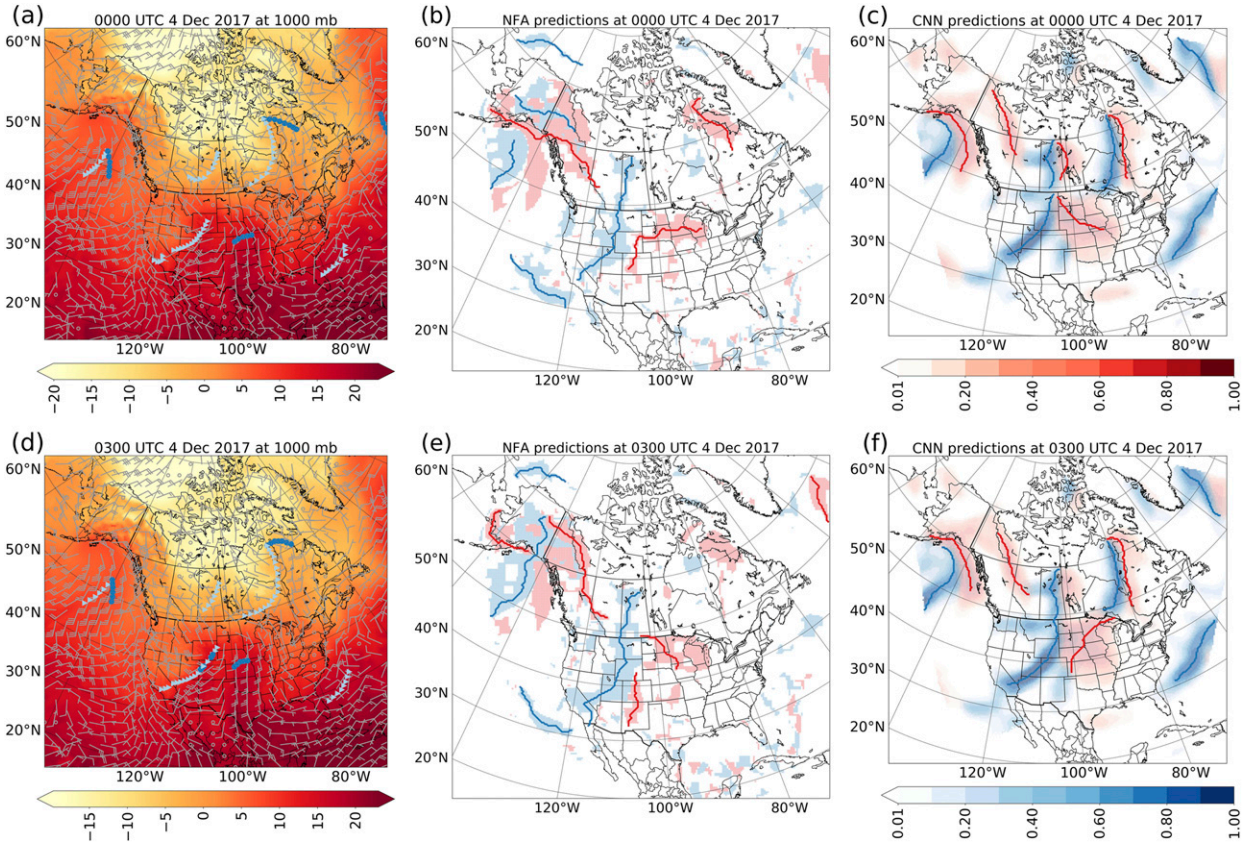

FIG. 15. As in Fig. 14, but for different time steps.

wider local context than just neighboring grid cells. When the CNN and WPC disagree, often the CNN prediction can be justified with reference to the predictors. For example, at 0000–0300 UTC (Fig. 14), the CNN and WPC disagree by several hundred kilometers on the cold front in the eastern United States. The predictors (wind shift and thermal gradients) support the CNN's frontal position, which is farther east. Surface fields (on which the WPC bulletins are explicitly based) are similar to the 1000-mb fields (cf. Figs. 5a,b and 5d,e).

The CNN and WPC also commonly disagree on short fronts. One example is the two WPC fronts in the eastern Pacific at 0000 UTC (Fig. 14), which the CNN considers one front. Also, the object conversion often produces fronts with unusual shapes, such as the very long cold front in the Atlantic (Fig. 14). Finally, the object conversion often produces fronts with unusual shapes, such as the very long cold front in the Atlantic from 0000to 0900 UTC and cold front in the Canadian prairies at 0900 UTC that mostly passes through areas of low gridded CF probability. Overall, the results suggest that object conversion is a more fruitful avenue for improvement than the CNN.

7. Summary and future work

We used convolutional neural nets (CNN) to identify warm and cold fronts. The predictors were small 1000-mb grids of u, v, T, q, θ_w, and/or Z. The target was the human label (no front, warm front, or cold front) at the center grid cell. We also developed a novel method to convert probability grids (the raw CNN output) to objects (polylines defining warm and cold fronts). We conducted experiments to find the best CNN parameters, the best object-conversion parameters, and the best numerical frontal analysis (NFA) method.

The outputs of Experiments 2 and 3—the best CNN and NFA methods with object conversion—were compared on testing data. The CNN dramatically outperformed NFA (Fig. 13). We used our own NFA method as the baseline because, to our knowledge, no previous work in NFA has published code or evaluated their method on more than a few examples. It is possible that another NFA method could perform comparably with our CNN. However, we believe that this study is sufficient to establish deep learning as a viable tool for front detection. To our knowledge, two previous studies have used deep learning (both CNNs) to

detect fronts. Racah et al. (2017) use an NFA method, rather than WPC bulletins, to create labels. Kunkel et al. (2018) use the WPC fronts as labels, but their presentation suggests that they match every predicted front with an actual front, regardless of distance. Thus, our results cannot be compared with either of these studies.

Our system could be used for many purposes, such as a spatial climatology of frontal occurrence and properties, or quantifying the spread in frontal properties across members of a numerical weather prediction (NWP) ensemble. With further development our system could also be used to evaluate NWP models with respect to frontal properties (e.g., identify biases in translation speed or strength). Future work will focus on improving the object-conversion method (e.g., the problems discussed in section 6b) and investigating the CNN's sensitivity to the source of labeled data (e.g., train with fronts drawn by meteorologists at another office). Also, we hope to try learning from 3D data and time series, which, although much more computationally expensive, could improve predictions further.

Acknowledgments. Funding was provided by the National Science Foundation (Grant EAGER NSF AGS 1802267) and NOAA/Office of Oceanic and Atmospheric Research (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 authors thank Drs. Jeffrey Basara, Jason Furtado, Michael Richman, and Andrew Fagg for helping to formulate the exam questions. We would also like to thank Alan Robson at the WPC for providing us with the archived bulletins.

APPENDIX

Performance Metrics

a. Performance metrics for gridcell-wise evaluation

Performance metrics for gridcell-wise evaluation are defined in the following equations:

$$\text{accuracy} = \frac{n_{11} + n_{22} + n_{33}}{N} = \frac{1}{N}\sum_{k=1}^{K} n_{kk}, \tag{A1}$$

$$\text{Heidke score} = \frac{\frac{1}{N}\sum_{k=1}^{K} n_{kk} - \frac{1}{N^2}\sum_{k=1}^{K} n(P_k)n(y_k)}{1 - \frac{1}{N^2}k = 1\sum_{k=1}^{K} n(P_k)n(y_k)}, \tag{A2}$$

$$\text{Peirce score} = \frac{\frac{1}{N}\sum_{k=1}^{K} n_{kk} - \frac{1}{N^2}\sum_{k=1}^{K} n(P_k)n(y_k)}{1 - \frac{1}{N^2}\sum_{k=1}^{K} n(y_k)^2}, \quad \text{and} \tag{A3}$$

$$\begin{cases} \text{Gerrity score} = \frac{1}{N}\sum_{i=1}^{K}\sum_{j=1}^{K} n_{ij}s_{ij}, \\ s_{ij} = s_{ji} = \frac{1}{K-1}\left[\sum_{k=1}^{i-1} a_k^{-1} - (j-i) + \sum_{k=j}^{K-1} a_k\right], \\ a_k = \frac{1 - \frac{1}{N}\sum_{r=1}^{k} n(y_r)}{\frac{1}{N}\sum_{r=1}^{k} n(y_r)}, \end{cases} \tag{A4}$$

where variable n_{ij} is the number of examples where the ith class is predicted and jth class is observed (Table 5); N is the total number of examples; $K = 3$ is the number of classes; $n(P_k)$ is the number of examples where the kth class is predicted; $n(y_k)$ is the number of examples where the kth class is observed; and $(1/N)\sum_{r=1}^{k} n(y_r)$, used to define a_k in Eq. (A4), is the cumulative observation frequency of the first k classes.

b. Performance metrics for object-based evaluation

Performance metrics for object-based evaluation are defined in the following equations:

$$\text{POD} = \frac{n_{\text{OTP}}}{n_{\text{OTP}} + n_{\text{FN}}}, \tag{A5}$$

$$\text{success ratio} = \text{SR} = \frac{n_{\text{PTP}}}{n_{\text{PTP}} + n_{\text{FP}}}, \tag{A6}$$

$$\text{frequency bias} = \frac{\text{POD}}{\text{SR}}, \quad \text{and} \tag{A7}$$

$$\text{CSI}^{-1} = \text{POD}^{-1} + \text{SR}^{-1} - 1, \tag{A8}$$

where n_{PTP} is the number of predicted fronts matched with an actual front (''prediction-oriented true positives''); n_{OTP} is the number of actual fronts matched with a predicted front (''observation-oriented true positives''); n_{FP} is the number of predicted fronts *not* matched with an actual front (false positives); and n_{FN} is the number of actual fronts *not* matched with a predicted front (false negatives). The matching is based on a neighborhood distance (section 4c). Equations (A5)–(A8) are used by the National Weather Service to verify tornado warnings (Brooks 2004, p. 1), another setting in

which negative examples (''nontornadoes'') are difficult to define.

REFERENCES

American Meteorological Society, 2014a: Equivalent temperature. Glossary of Meteorology, accessed 13 March 2018, http://glossary.ametsoc.org/wiki/Equivalent_temperature.

——, 2014b: Front. Glossary of Meteorology, accessed 13 March 2018, http://glossary.ametsoc.org/wiki/Front.

——, 2014c: Wet-bulb potential temperature. Glossary of Meteorology, accessed 13 March 2018, http://glossary.ametsoc.org/wiki/Pseudo_wet-bulb_potential_temperature.

——, 2014d: Hypsometric equation. Glossary of Meteorology, accessed 13 March 2018, http://glossary.ametsoc.org/wiki/Hypsometric_equation.

Baldi, P., and P. Sadowski, 2013: Understanding dropout. *Advances in Neural Information Processing Systems*, Vol. 26, Lake Tahoe, NV, Neural Information Processing Systems, https://papers.nips.cc/paper/4878-understanding-dropout.

Berry, G., M. Reeder, and C. Jakob, 2011: A global climatology of atmospheric fronts. *Geophys. Res. Lett.*, **38**, L04809, https://doi.org/10.1029/2010GL046451.

Brooks, H., 2004: Tornado-warning performance in the past and future: A perspective from signal detection theory. *Bull. Amer. Meteor. Soc.*, **85**, 837–843, https://doi.org/10.1175/BAMS-85-6-837.

Catto, J., and S. Pfahl, 2013: The importance of fronts for extreme precipitation. *J. Geophys. Res.*, **118**, 10 791–10 801, https://doi.org/10.1002/jgrd.50852.

Chollet, F., 2017: Xception: Deep learning with depthwise separable convolutions. *IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, Institute of Electrical and Electronics Engineers (IEEE), https://doi.org/10.1109/CVPR.2017.195.

Clarke, L., and R. Renard, 1966: The U.S. Navy numerical frontal analysis scheme: Further development and a limited evaluation. *J. Appl. Meteor.*, **5**, 764–777, https://doi.org/10.1175/1520-0450(1966)005<0764:TUSNNF>2.0.CO;2.

Efron, B., 1979: Bootstrap methods: Another look at the jackknife. *Ann. Stat.*, **7**, 1–26, https://doi.org/10.1214/aos/1176344552.

Fawbush, E., and R. Miller, 1954: The types of airmasses in which North American tornadoes form. *Bull. Amer. Meteor. Soc.*, **35**, 154–165, https://doi.org/10.1175/1520-0477-35.4.154.

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., II, S. Haupt, D. Nychka, and G. Thompson, 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.

Gil, Y., and Coauthors, 2019: Intelligent systems for geosciences: An essential research agenda. *Commun. ACM*, **62**, 76–84, https://doi.org/10.1145/3192335.

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

Hewson, E., 1936: The application of wet-bulb potential temperature to air mass analysis. *Quart. J. Roy. Meteor. Soc.*, **62**, 387–420, https://doi.org/10.1002/qj.49706226604.

Hewson, T., 1998: Objective fronts. *Meteor. Appl.*, **5**, 37–65, https://doi.org/10.1017/S1350482798000553.

Hinton, G., N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2012: Improving neural networks by preventing co-adaptation of feature detectors. arXiv: 1207.0580v1.

Holton, J., 2004: *An Introduction to Dynamic Meteorology*. 4th ed. Academic Press, 535 pp.

Huber-Pock, F., and C. Kress, 1981: Contributions to the problem of numerical frontal analysis. *Proc. Symp. on Current Problems of Weather Prediction*, Vienna, Austria, Zentralanstalt für Meteorologie und Geodynamik, 51–55.

Ioffe, S., and C. Szegedy, 2015: Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Int. Conf. on Machine Learning*, Lille, France, International Machine Learning Society, http://proceedings.mlr.press/v37/ioffe15.pdf.

Jenkner, J., M. Sprenger, I. Schwenk, C. Schwierz, S. Dierer, and D. Leuenberger, 2010: Detection and climatology of fronts in a high-resolution model reanalysis over the Alps. *Meteor. Appl.*, **17** (1), 1–18.

Krause, J., B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei, 2016: The unreasonable effectiveness of noisy data for fine-grained recognition. *European Conference on Computer Vision—ECCV 2016*, B. Leibe et al., Eds., Springer, https://doi.org/10.1007/978-3-319-46487-9_19.

Krizhevsky, A., I. Sutskever, and G. Hinton, 2012: ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, Lake Tahoe, NV, Neural Information Processing Systems, https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

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.

Kurth, T., and Coauthors, 2018: Exascale deep learning for climate analytics. *Int. Conf. for High Performance Computing, Networking, Storage, and Analysis*, Dallas, TX, Institute of Electrical and Electronics Engineers (IEEE), Article No. 51.

Low, T., and D. Hudak, 1997: Development of air mass climatology analysis for the determination of characteristic marine atmospheres. Part I: North Atlantic. *Theor. Appl. Climatol.*, **57**, 135–153, https://doi.org/10.1007/BF00863609.

Mesinger, F., and Coauthors, 2006: North American Regional Reanalysis. *Bull. Amer. Meteor. Soc.*, **87**, 343–360, https://doi.org/10.1175/BAMS-87-3-343.

Miller, R., 1959: Tornado-producing synoptic patterns. *Bull. Amer. Meteor. Soc.*, **40**, 465–472, https://doi.org/10.1175/1520-0477-40.9.465.

Mitchell, T., 1997: *Machine Learning*. 1st ed. McGraw Hill, 432 pp.

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.

National Weather Service, 2007: Reading the coded surface bulletin. NOAA/NWS, accessed 13 March 2018, http://www.wpc.ncep.noaa.gov/html/read_coded_bull.shtml.

Racah, E., C. Beckham, T. Maharaj, S. Kahou, Prabhat, and C. Pal, 2017: ExtremeWeather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. *31st Conf. on Neural Information*

Processing Systems (NIPS 2017), Long Beach, CA, Neural Information Processing Systems, https://papers.nips.cc/paper/6932-extremeweather-a-large-scale-climate-dataset-for-semi-supervised-detection-localization-and-understanding-of-extreme-weather-events.pdf.

Rasp, S., M. Pritchard, and P. Gentine, 2018: Deep learning to represent subgrid processes in climate models. *Proc. Natl. Acad. Sci. USA*, **115**, 9684–9689, https://doi.org/10.1073/pnas.1810286115.

Ravuri, M., A. Kannan, G. Tso, and X. Amatriain, 2018: Learning from the experts: From expert systems to machine-learned diagnosis models. *Proc. Mach. Learning Res.*, **85**, 1–16.

Reichstein, M., G. Camps-Balls, B. Stevens, M. Jung, J. Denzler, and N. Carvalhais, 2019: Deep learning and process understanding for data-driven Earth system science. *Nature*, **566**, 195–204, https://doi.org/10.1038/s41586-019-0912-1.

Renard, R., and L. Clarke, 1965: Experiments in numerical objective frontal analysis. *Mon. Wea. Rev.*, **93**, 547–556, https://doi.org/10.1175/1520-0493(1965)093<0547:EINOFA>2.3.CO;2.

Roebber, P., 2009: Visualizing multiple measures of forecast quality. *Wea. Forecasting*, **24**, 601–608, https://doi.org/10.1175/2008WAF2222159.1.

Roeder, W., and R. Gall, 1987: Estimating the width of a typical cold front. *Natl. Wea. Dig.*, **12** (4), 17–19.

Russell, S., and P. Norvig, 2010: *Artificial Intelligence: A Modern Approach*. 3rd ed. Pearson, 1152 pp.

Schemm, S., I. Rudeva, and I. Simmonds, 2015: Extratropical fronts in the lower troposphere—Global perspectives obtained from two automated methods. *Quart. J. Roy. Meteor. Soc.*, **141**, 1686–1698, https://doi.org/10.1002/qj.2471.

Scher, S., 2018: Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning. *Geophys. Res. Lett.*, **45**, 12 616–12 622, https://doi.org/10.1029/2018GL080704.

Serreze, M., A. Lynch, and M. Clark, 2001: The Arctic frontal zone as seen in the NCEP–NCAR reanalysis. *J. Climate*, **14**, 1550–1567, https://doi.org/10.1175/1520-0442(2001)014<1550:TAFZAS>2.0.CO;2.

——, A. Barrett, A. Slater, M. Steele, J. Zhang, and K. Trenberth, 2007: The large-scale energy budget of the Arctic. *J. Geophys. Res.*, **112**, D11122, https://doi.org/10.1029/2006JD008230.

Simmonds, I., K. Keay, and J. Bye, 2012: Identification and climatology of Southern Hemisphere mobile fronts in a modern reanalysis. *J. Climate*, **25**, 1945–1962, https://doi.org/10.1175/JCLI-D-11-00100.1.

Wang, L., K. Scott, L. Xu, and D. Clausi, 2016: Sea ice concentration estimation during melt from dual-pol SAR scenes using deep convolutional neural networks: A case study. *IEEE Trans. Geosci. Remote Sens.*, **54**, 4524–4533, https://doi.org/10.1109/TGRS.2016.2543660.

Wimmers, A., C. Velden, and J. Cossuth, 2019: Using deep learning to estimate tropical cyclone intensity from satellite passive microwave imagery. *Mon. Wea. Rev.*, **147**, 2261–2282, https://doi.org/10.1175/MWR-D-18-0391.1.