Using Deep Learning to Nowcast the Spatial Coverage of Convection from *Himawari-8* Satellite Data

RYAN LAGERQUIST,[a] JEBB Q. STEWART,[b] IMME EBERT-UPHOFF,[c,d] AND CHRISTINA KUMLER[e]

[a] *Cooperative Institute for Research in the Atmosphere, National Oceanic and Atmospheric Administration/Earth System Research Laboratory/Global Systems Laboratory, Boulder, Colorado*
[b] *NOAA/ESRL/GSL, Boulder, Colorado*
[c] *Cooperative Institute for Research in the Atmosphere, Colorado State University, Fort Collins, Colorado*
[d] *Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, Colorado*
[e] *Cooperative Institute for Research in Environmental Sciences, NOAA/ESRL/GSL, Boulder, Colorado*

(Manuscript received 28 April 2021, in final form 15 September 2021)

ABSTRACT: Predicting the timing and location of thunderstorms (''convection'') allows for preventive actions that can save both lives and property. We have applied U-nets, a deep-learning-based type of neural network, to forecast convection on a grid at lead times up to 120 min. The goal is to make skillful forecasts with only present and past satellite data as predictors. Specifically, predictors are multispectral brightness-temperature images from the *Himawari-8* satellite, while targets (ground truth) are provided by weather radars in Taiwan. U-nets are becoming popular in atmospheric science due to their advantages for gridded prediction. Furthermore, we use three novel approaches to advance U-nets in atmospheric science. First, we compare three architectures—vanilla, temporal, and U-net++—and find that vanilla U-nets are best for this task. Second, we train U-nets with the fractions skill score, which is spatially aware, as the loss function. Third, because we do not have adequate ground truth over the full *Himawari-8* domain, we train the U-nets with small radar-centered patches, then apply trained U-nets to the full domain. Also, we find that the best predictions are given by U-nets trained with satellite data from multiple lag times, not only the present. We evaluate U-nets in detail—by time of day, month, and geographic location—and compare them to persistence models. The U-nets outperform persistence at lead times $\geq$ 60 min, and at all lead times the U-nets provide a more realistic climatology than persistence. Our code is available publicly.

KEYWORDS: Deep convection; Satellite observations; Time series; Nowcasting; Deep learning; Neural networks

1. Introduction

Thunderstorms (hereafter ''convection'') are a dangerous weather phenomenon, causing economic losses, injury, and death. Convection heavily impacts industries such as aviation, outdoor events, and wind energy. In these and other activities, predicting the location and timing of convection, even at short lead times, allows for preventive actions that mitigate both human and economic losses (Wilson and Mueller 1993; Mueller et al. 1993; Ahijevych et al. 2016). In Taiwan especially, hazards commonly associated with convection are flash flooding and landslides, due to the country's steep terrain, high rainfall rates, and frequent earthquakes that weaken slopes (Lin et al. 2017). Forecasting at lead times $\leq$ 3 h is often called nowcasting, and much work has been done on nowcasting the location and timing of convection. Early work used primarily radar data for this purpose, while more recent work has used primarily satellite data, due to the increased spatial and temporal resolution of geostationary satellites over time. Also, satellites can detect thunderstorms earlier in their development (i.e., before they develop enough precipitation to produce a radar echo), and satellites cover a much larger portion of the globe than do radars.

To our knowledge, Mueller and Wilson (1989) developed the first explicit convection-forecasting algorithm. They used the radar at Denver International Airport (DIA) to detect lines of boundary layer convergence, then used properties of these lines to forecast the probability of radar reflectivity $>$ 30 dBZ at lead times up to 3 h. Wilson and Mueller (1993) expanded on this work, forecasting convection over an 8000-km^2 area surrounding DIA, but at lead times $\leq$ 30 min. Neither study used satellite data, because at this time geostationary satellites had a temporal resolution of 30 min, deemed too coarse to be useful (Wilson and Mueller 1993). Mueller et al. (1993) investigated the potential of high-resolution surface (10–15-km spacing and 1-min time steps) and sounding (8 sites over 25 000 km^2 and 1–6-h time steps) observations to forecast convective initiation (CI), but they found that the high-resolution data provide no skill beyond routine observations (25–50-km spacing for surface stations and one morning sounding). This conclusion is important, as modern surface and sounding networks still do not have high resolution as defined in Mueller et al. (1993). Mueller et al. (1993) suggested that CI-forecasting could instead be improved by using high-resolution satellite data.

To our knowledge, Roberts and Rutledge (2003) developed the first explicit convection-forecasting algorithm based on satellite data. They forecast CI at lead times up to 1 h, primarily

Supplemental information related to this paper is available at the Journals Online website: https://doi.org/10.1175/MWR-D-21-0096.s1.

Corresponding author: Ryan Lagerquist, ralager@colostate.edu

DOI: 10.1175/MWR-D-21-0096.1

near DIA, and outperformed earlier methods. Mecikalski and Bedka (2006, hereafter MB06) developed a CI-forecasting algorithm called Satellite Convection Analysis and Tracking (SATCAST). This expert system used eight "interest fields" (predictors), all based on infrared GOES data, to forecast at lead times up to 1 h. SATCAST improved upon earlier work by using multispectral infrared data, including band differences (e.g., 13.3-μm minus 10.7-μm brightness temperature) and temporal changes thereof. MB06 is still a highly influential study, as the interest fields developed therein have been used extensively since. Mecikalski et al. (2008) provided an objective evaluation of SATCAST, finding that it had a very high probability of detection (POD; 0.99) but also a very high false-alarm ratio[1] (FAR; 0.72). Mecikalski et al. (2008) is also a highly influential study, as it established that CI-forecasting algorithms tend to have a high FAR, which much work since has focused on reducing.

Sieglaff et al. (2011) developed the University of Wisconsin Convective Initiation (UWCI) algorithm, which forecast CI at lead times up to 1 h. The UWCI improved upon earlier methods by using a box-averaging approach, rather than explicit cumulus-tracking, to compute temporal changes in infrared brightness temperature. Explicit tracking, as in SATCAST, was error-prone at the time, due partly to the long time interval (15 min for GOES) between consecutive satellite images. Otherwise, the UWCI was similar to previous algorithms such as the Autonowcast System (Mueller et al. 2003) and SATCAST – an expert system based on infrared data. Walker et al. (2012) developed SATCASTv2, which improved upon the UWCI by reintroducing explicit cumulus-tracking. Despite its advantages, SATCASTv2 was a daytime-only algorithm with high FAR — e.g., 0.55 in the central United States, where it performed best overall. Mecikalski et al. (2015) developed the GOES-R CI algorithm, which had two advantages over earlier methods. First, it used machine learning (ML; specifically logistic regression or a random forest), allowing for probabilistic, rather than binary, forecasts. Second, the predictors included NWP data [from the Rapid Refresh (RAP) model], allowing for a dramatic decrease in FAR. Mecikalski et al. (2015) found that the two most important predictors were surface-based and most unstable convective inhibition (CIN), followed by surface-based and most unstable convective available potential energy (CAPE), all derived from NWP.

Lee et al. (2017) developed a convection-forecasting algorithm for the *Himawari-8* satellite, which covers the western Pacific and eastern Asia. They used random forests with 12 infrared-based interest fields as predictors, similar to those used in SATCAST. Han et al. (2019) expanded on this work by using a procedure to iteratively expand the training set for the random forest — at each step, adding cases similar to those that the random forest predicts poorly. Although most of their evaluation scores were worse than in Lee et al. (2017), the random forest of Han et al. (2019) detected incipient convection at an earlier stage of development and thus had a longer lead time. Lee et al. (2021) developed a convolutional neural network (CNN), a type of deep-learning method, to detect convection at the present time. This followed previous work using deep learning to forecast precipitation amount (Shi et al. 2015; Sønderby et al. 2020). Other than the use of deep learning, a major advantage of Lee et al. (2021) is that they defined convection by a sophisticated echo-classification algorithm, which incorporated more radar-measured information than a simple reflectivity threshold.

TABLE 1. Characteristics of *Himawari-8* spectral bands used to create predictors. All bands listed have a spatiotemporal resolution of 10 min and 2 km.

Band No.	Central wavelength (μm)	Bandwidth (μm)
8	6.25	0.37
9	6.95	0.12
10	7.35	0.17
11	8.60	0.32
13	10.45	0.30
14	11.20	0.20
16	13.30	0.20

We apply neural networks, an ML method, to forecast convection at lead times up to 2 h. The predictors are a time series of brightness-temperature grids from seven infrared bands on the *Himawari-8* satellite,[2] and the output is a grid of convection probabilities at the given lead time.[3] The labels (treated as correct answers) are produced by applying an echo-classification algorithm to data from weather radars in Taiwan. Four characteristics of our work make it unique from previous work. First, we use U-nets (Ronneberger et al. 2015), which are similar to CNNs but better suited for pixelwise prediction (here, predicting the convection probability at each grid point). U-nets are quickly gaining popularity in atmospheric science (Chen et al. 2021; Kumler-Bonfanti et al. 2020; Sadeghi et al. 2020; Sha et al. 2020a, b; Lagerquist et al. 2021), and herein we use them for another pixelwise prediction task, to forecast convection on a grid. Second, we experiment with two novel U-net architectures—the temporal U-net and U-net++—as well as the vanilla U-net. Third, we use a sliding-window approach, allowing us to train U-nets with small patches of the full grid (those with adequate radar coverage to create labels), then apply U-nets to the full grid at inference time. Fourth, we use a spatially aware function, called the fractions skill score (FSS; Roberts and Lean 2008), as the loss function. The FSS does not unduly punish small offsets between forecast and observed convection; it has been used widely in atmospheric science (Mittermaier 2021), but to our

[1] Not to be confused with false-alarm *rate*, often called the probability of false detection (POFD). POFD is $b/(b + d)$, and FAR is $b/(a + b)$, where a is the number of true positives, b is the number of false positives, and d is the number of true negatives.

[2] The goal is to obtain skillful forecasts with only satellite-based predictors, per request of the Taiwan Central Weather Bureau, the direct beneficiaries of this project.

[3] Technically, a classification model outputs confidence scores rather than calibrated probabilities. However, for the sake of convenience, we refer to the confidence scores, which range continuously from [0, 1], as probabilities.

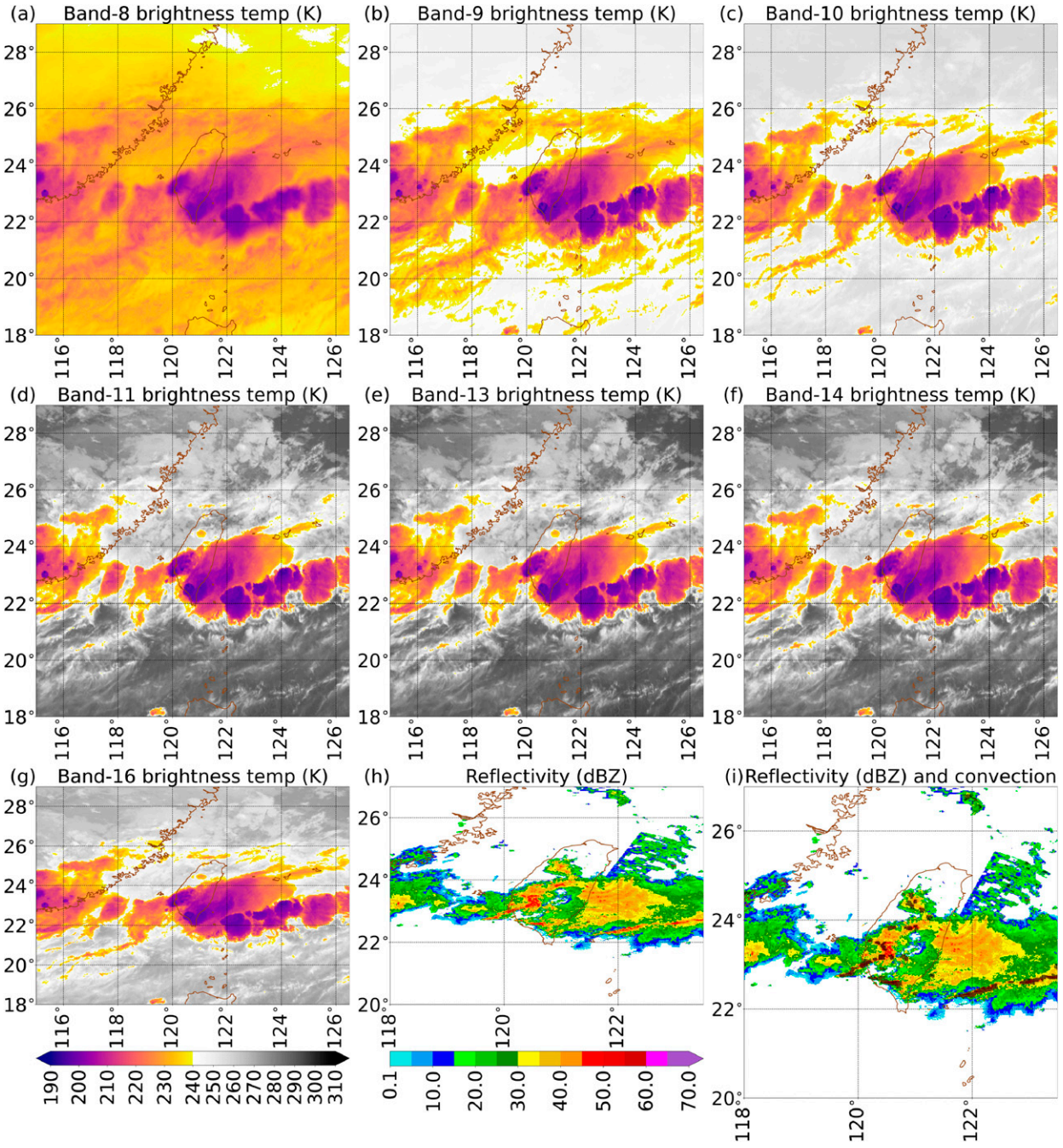

FIG. 1. Input data valid at 1800 UTC 3 Jun 2017. (a)–(g) Brightness temperature (K) in each spectral band, used as predictors; see the color bar below (g). Composite (column-maximum) radar reflectivity (h) with and (i) without echo classification. The black dots in (i) show grid points with convection, according to SL3D (section 2c).

knowledge it has been used only for post hoc evaluation, never as the loss function for training a model.

The rest of this paper is organized as follows. Section 2 describes the input data and preprocessing; section 3 describes the U-net architectures attempted; section 4 details the ML methodology; Sections 5 and 6 evaluate and interpret the final U-net models; and section 7 provides a summary and list of future work.

2. Input data

a. Data description

The predictors come from *Himawari-8* satellite data, provided by the Taiwan Central Weather Bureau (CWB) at 10-min time steps for 3 years: 2016–18. The data consist of gridded radiance maps for seven spectral bands, listed in Table 1.

The weighting function for each band is shown in Fig. 1 of Da (2015). All seven bands are in the infrared part of the spectrum, so the same data can be used during both day and night. We convert the radiances to brightness temperatures, using lookup tables provided by the CWB. Because brightness temperature has traditionally been used to forecast convection (section 1), we find that it is more interpretable than radiance. The satellite data are provided on a grid spanning 18–29°N, 115–126.5°E with 0.0125° spacing, as shown in Figs. 1a–g.

The labels come from radar data (Figs. 1h–i), also provided by the CWB at 10-min time steps for the years 2016–18. The files provided contain reflectivity on a 3D grid created by the CWB, who interpolated data from the four weather radars in Taiwan (Chang et al. 2009), all S-band (10-cm wavelength), to a common grid. The radar domain is a subset of the satellite domain: 20°–27°N, 118°–123.5°E, also with 0.0125° spacing. Heights in the grid range from 0 to 17 km above sea level (MSL), with 0.5-km spacing up to 5 km MSL and 1-km spacing aloft.

Both the satellite and radar data have 10-min time steps, and we make a prediction for each time step. The training, validation, and testing data are split by year as shown in Table 2. With no missing data, there would be 51 696 examples in the training period (359 days × 144 time steps per day, 51 552 examples in the validation period (358 days), and 52 560 examples in the testing period. However, at each initial time t_0, we make predictions only if:

(i) The satellite data, used as predictors, are available at all required lag times (e.g., t_0, $t_0 - 10$ min, and $t_0 - 20$ min);
(ii) The radar data, used as labels, are available at the required lead time (e.g., $t_0 + 60$ min).

Thus, the number of available examples depends on the lag times and lead time required by the U-net. In general, U-nets with more lag times have fewer available examples (with more lag times, there is a greater chance that at least one lag time is missing). Typically, the number of available examples is ~90% of the possible total. For instance, for a U-net with lag times of {0, 20, 40} and lead time of 60 min, the number of available examples is 45 945 for the validation data (89.1% of possible) and 48 774 for the testing data (92.8% of possible).

b. Preprocessing of satellite data

During this research, we discovered that linear artifacts are common in band 8 (wavelength of 6.25 μm). We developed a quality-control algorithm to remove these artifacts, described below for one time step. See Fig. S1 in the online supplemental material for an example of the result.

1) Create a map of smoothed brightness temperatures, using a 5 × 5 mean filter. Let raw and smoothed brightness temperature be T_b and T_b', respectively.
2) At each grid point, compute the absolute difference: $\Delta T_b = |T_b - T_b'|$. A grid point with large ΔT_b varies strongly from its neighbors.

TABLE 2. Training, validation, and testing data. Valid time is the time at which the prediction is valid. For example, if the forecast-issue time is 1100 UTC and the lead time is 60 min, the valid time is 1200 UTC. There is a 1-week gap between each pair of consecutive datasets to eliminate temporal autocorrelation and ensure that the three sets are truly independent.

Dataset	Valid times
Training	1 Jan–24 Dec 2016
Validation	1 Jan–24 Dec 2017
Testing	1 Jan–31 Dec 2018

3) Dilate the ΔT_b map, using a 5 × 5 maximum filter and letting the result be $\widetilde{\Delta T_b}$. The purpose is to fill "holes" (grid points with small ΔT_b surrounded by neighbors with large ΔT_b).
4) Find connected regions[4] of at least 1000 grid points where $\widetilde{\Delta T_b} > 1$ K. Call these "flagged regions"; they do not naturally occur without erroneous data.
5) At each grid point in a flagged region, replace T_b with the linearly interpolated T_b from grid points outside all flagged regions.

Due to the dilation in step 3, flagged regions contain both erroneous and some nonerroneous grid points. However, we find this trade-off acceptable, because if linear artifacts are not removed the U-nets almost always interpret the associated large temperature gradients as convection.

c. Preprocessing of radar data

To create convection masks (labels), we apply an echo-classification algorithm to the radar data. The algorithm is called Storm-labeling in 3 Dimensions (SL3D; Starzec et al. 2017) and labels each horizontal grid location as convective or nonconvective. For example, in Fig. 1i each convective location is marked with a black dot. Lagerquist et al. (2020) modified SL3D for tornado prediction, as described in their supplemental material. We have made one more modification to the version described in Lagerquist et al. (2020): each convective grid point must be in a connected region of ≥10 convective grid points. To achieve this, we have added a final step called the region filter: for any convective grid point not in a connected region of ≥10 convective grid points, we change the label to nonconvective. We have found the region filter necessary to remove areas of high-reflectivity but nonmeteorological echoes (e.g., ground clutter), which tend to be larger in the Taiwan data than in the U.S. data for which SL3D was originally developed. The disadvantage of the region filter is that it removes early- and late-stage thunderstorms, in which <10 grid points meet the other SL3D criteria. On the radar grid used, a connected region of <10 grid points has an area no greater than 16.34 km^2.

[4] A connected region is a set R of grid cells, such that each grid cell in R shares an edge or corner with another grid cell in R.

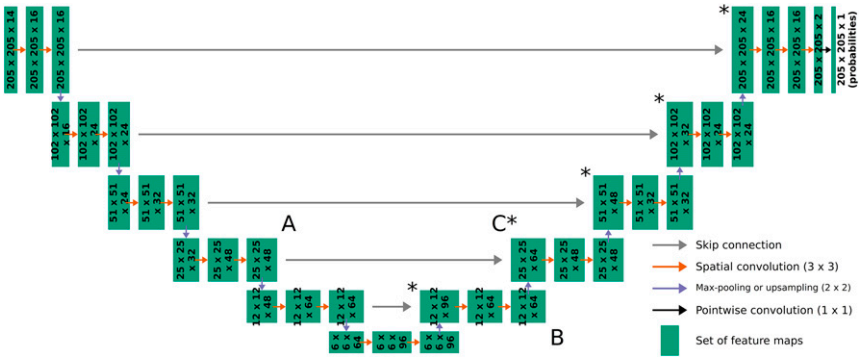

FIG. 2. Architecture of vanilla U-net with two lag times for predictors. The left side of the U-shape is the downsampling side; the right side is the upsampling side; and depth increases from the input layer in the top-left corner to the output layer in the top-right corner. In each set of feature maps, the numbers are dimensions (rows × columns × channels). The 14 input channels are the predictor variables—i.e., brightness-temperature maps from 7 spectral bands at 2 lag times. Spatial-convolution filters have dimensions of 3 rows × 3 columns; pointwise-convolution filters have dimensions of 1 row × 1 column; while pooling and upsampling windows have dimensions of 2 rows × 2 columns. For each set of feature maps with two incoming arrows (upsampling and skip connection), there is an extra convolutional layer (with 3 × 3 filters) that reduces the number of channels. For example, stacking the feature maps A with the upsampled version of feature maps B results in 48 + 64 = 112 channels. The extra convolutional layer here transforms the 112 channels to the 64 channels in C.

3. U-net architectures

a. Background on U-nets

A traditional neural network, or fully connected neural network (FCNN; chapter 6 of Goodfellow et al. 2016), contains several layers of neurons. Each neuron performs its own linear regression, with weights learned during training, and each layer of neurons is followed by a nonlinear activation function, such as the hyperbolic tangent or sigmoid. The main disadvantage of FCNNs is that they treat all predictors as independent scalars, making them unable to exploit spatial and temporal structure. Convolutional neural networks (CNN; Fukushima 1980; Fukushima and Miyake 1982) use convolutional filters to detect spatial and temporal features in gridded data, thus overcoming the disadvantage of FCNNs. U-nets (Ronneberger et al. 2015) retain this advantage of CNNs and also excel at pixelwise prediction—i.e., making a prediction at every grid point—due to their use of skip connections, discussed in section 3b. We experiment with three U-net architectures, explained briefly below and in detail in the original papers.

b. Vanilla U-net

A vanilla U-net (Ronneberger et al. 2015) contains four types of components, shown in Fig. 2: convolutional layers, pooling (downsampling) layers, upsampling layers, and skip connections. The left side of the U-shape is the downsampling side, where spatial[5] resolution decreases with depth, and the right side is the upsampling side, where spatial resolution increases with depth. The convolutional layers detect spatial features, and the other components allow different convolutional layers to detect features at different resolutions. This is crucial for weather prediction, due to the multiscale nature of weather phenomena. Inputs to the first layer are raw predictors, and inputs to deeper layers[6] are transformed versions of the raw predictors, called feature maps. Convolution is both a spatial and multivariate transformation, so the feature maps encode spatial patterns that include all predictor variables. In a CNN or U-net, a nonlinear activation function comes after each convolutional layer. The inner workings of a convolutional layer are animated in supplemental Fig. S1 of Lagerquist et al. (2020).

Each pooling layer downsamples the feature maps to a lower spatial resolution, typically using a 2 × 2 maximum filter. Thus, on the downsampling side of Fig. 2, grid spacing increases from 0.0125° to 0.025°, 0.05°, 0.1°, 0.2°, and finally 0.4° at the bottom. As the spatial resolution decreases, the number of feature maps (''channels'') typically increases, to offset the loss of spatial information. The inner workings of a pooling layer are animated in supplemental Fig. S2 of Lagerquist et al. (2020).

Each upsampling layer upsamples the feature maps to a higher spatial resolution, using interpolation followed by convolution. The convolution is crucial because interpolation alone cannot adequately reconstruct high-resolution information from low-resolution information. As the spatial resolution

[5] For U-nets that perform spatiotemporal convolution, replace ''spatial'' with ''spatiotemporal'' in this and the next three paragraphs. Our vanilla U-nets perform only spatial convolution for reasons explained at the end of this subsection.

[6] Depth increases while traveling from the input layer, at the top left of the U, to the output layer, at the top right of the U.

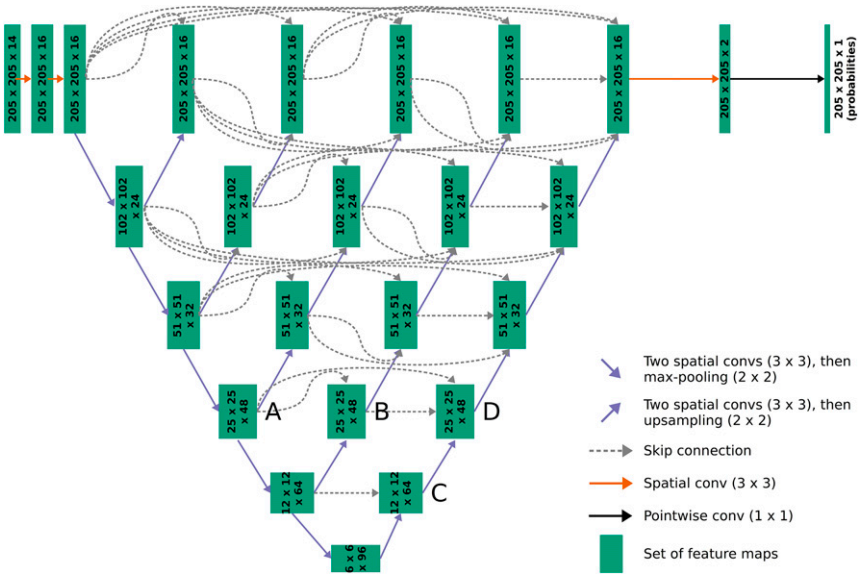

FIG. 3. Architecture of U-net++ with two lag times for predictors. Each "downsampling" arrow corresponds to a pooling layer followed by two convolutional layers, as in one row of the downsampling side in Fig. 2. Each "upsampling" arrow corresponds to an upsampling layer followed by two convolutional layers, as in one row of the upsampling side in Fig. 2. For each set of feature maps with multiple incoming arrows (upsampling and skip connections), there is an extra convolutional layer (with 3×3 filters) that reduces the number of channels. For example, stacking the feature maps A with B and the upsampled version of feature maps C results in $48 + 48 + 64 = 160$ channels. The extra convolutional layer here transforms the 160 channels to the 64 channels in D.

increases, the number of channels typically decreases, terminating with the number of output channels (here, one: convection probability).

Skip connections preserve high-resolution information from the downsampling side and carry it to the upsampling side. Without skip connections, the U-net would simply perform downsampling and then upsampling, which is a lossy operation. On the upsampling (right) side of Fig. 2, each feature-map set labeled with an asterisk (*) is formed by concatenating feature maps from the upsampling layer below and the skip connection to the left. Although both incoming feature-map sets have the same nominal spatial resolution, those from the skip connection have a higher effective resolution, because less information therein has been lost by downsampling. Feature maps from the upsampling layer have two advantages: (i) they contain higher-level abstractions, because they have passed through more convolutions and nonlinear activations; (ii) they contain wider spatial context, because they are upsampled from coarser resolution.

In the vanilla architecture, we concatenate predictors (brightness-temperature maps) from different lag times along the channel dimension, so different spectral channels and different lag times are treated equivalently. In principle, it is possible to reserve one dimension for lag times and one for spectral channels (so the inputs in Fig. 2 would be $205 \times 205 \times 2 \times 7$, rather than $205 \times 205 \times 14$), then perform spatiotemporal convolution rather than spatial convolution. However, spatiotemporal convolution is computationally expensive (i.e., 3D convolution is much more expensive than 2D convolution), and in our experience with past projects, it does not lead to better performance.

c. *U-net++*

A U-net++ (Zhou et al. 2019) contains more skip connections than a vanilla U-net, allowing features from more than two spatial scales to be combined at each level (Fig. 3). For example, the feature-map set labeled D in Fig. 3 is formed by concatenating A, B, and the upsampled version of C. Although the feature maps all have a nominal resolution of 0.1°, their effective resolutions, due to downsampling, are 0.1°, 0.2°, and 0.4°, respectively. By having more skip connections, the U-net++ allows information to flow along the most useful paths, causing it to outperform the vanilla U-net for some tasks (Zhou et al. 2019).

In the U-net++ architecture, as in the vanilla architecture, channels and lag times are treated equivalently.

d. *Temporal U-net*

A temporal U-net (Chiu et al. 2020) is similar to a vanilla U-net, except that it processes each lag time independently on the downsampling side (left in Fig. 4), then combines features from the different lag times via spatiotemporal convolution (middle of Fig. 4). The spatiotemporal-convolution layers are called the "temporal forecasting module" in Chiu et al. (2020). Thus, the temporal U-net, unlike the vanilla and U-net++ architectures, treats channels and lag times differently. Chiu et al. (2020) found that this ability allows the temporal U-net to

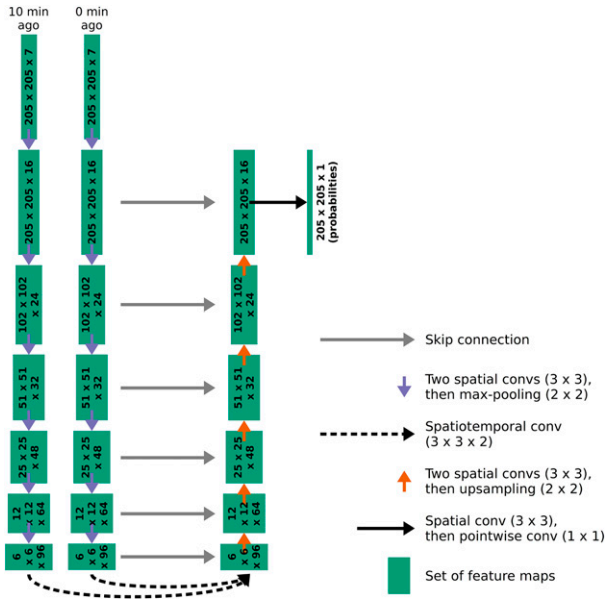

FIG. 4. Architecture of temporal U-net with lag times of 0 and 10 min for predictors. As in Fig. 2, the left side of the U-shape is the downsampling side; the right side is the upsampling side; and depth increases from the input layer in the top-left corner to the output layer in the top-right corner. At each lag time, the 7 channels correspond to the 7 spectral bands of the satellite. As in Figs. 2 and 3 for each set of feature maps with two incoming arrows (upsampling and skip connection), there is an extra convolutional layer (with 3×3 filters) that reduces the number of channels.

outperform the vanilla U-net, and we hypothesize that for our task, said ability will allow the temporal U-net to outperform both the vanilla and U-net++ architectures.

4. Machine-learning methodology

This section describes our ML methodology, other than the tuning of hyperparameters, which is discussed in the supplemental material. Hyperparameters are model settings not optimized by training, such as the U-net architecture and lag times for predictors, which are two of the four hyperparameters that we tune. We split the data into training, validation, and testing sets (Table 2), using only the validation set to tune hyperparameters. We use multiple scores to select the best model at each lead time, as discussed in the supplemental material. The main results of hyperparameter-tuning are (i) the vanilla U-net is the best architecture, contrary to our stated hypothesis in section 3d; (ii) regardless of lead time, the best performance is achieved with multiple lag times for predictors. In other words, the temporal evolution of satellite images is important as expected, but surprisingly, the best U-net architecture is the simplest. In an earlier experiment (not shown), before omitting the northernmost radar (for reasons explained in section 4a), we found that the vanilla U-net was the *worst* architecture, based on validation data. This suggests that the more complex architectures (temporal and U-net++) fit systematic errors from the northernmost radar more strongly.

a. Training with patches

We train each U-net with small radar-centered patches, rather than the full grid. This avoids issues with limited radar coverage and memory constraints.[7] As shown in Fig. 5a, where distance from the nearest radar (d_{nr}) >100 km, artifacts in convection frequency are more severe than where $d_{nr} \leq 100$ km. Also, we have found that data from the northernmost radar (located at the northern tip of Taiwan, not circled in Fig. 5a) contain a large number of severe errors, so we do not use data from this radar. The radar-centered patches have a complete domain of 205×205 grid points (2.5625° × 2.5625°) and inner domain of 105×105 grid points (1.3125° × 1.3125°), as shown in Fig. 5b. Each U-net reads predictors from the complete domain (to avoid edge effects) but makes predictions only for the inner domain, where there is adequate radar coverage.

b. Inference with sliding windows

At inference time (i.e., when using a trained U-net to predict the full grid), we use the sliding-window approach shown in Fig. 5c, similar to Liu et al. (2018). Specifically, we slide the 205×205 window by 25 grid points at a time, leading to a large overlap between adjacent inner windows. We apply the U-net to each window, ignoring predictions (convection probabilities) in the outer domain. Due to the large overlap between adjacent inner windows, most grid points receive more than one prediction. At these grid points we average all the predictions. Despite this averaging, there are sometimes sharp gradients in the probability map. To remove sharp gradients, we apply a Gaussian smoother to the final probability map, with an e-folding radius of two grid points.

c. Loss function

The loss function, used to optimize U-net weights during training, is the fractions skill score (FSS; Roberts and Lean 2008). For one example (i.e., the actual and forecast convection at one time step), the FSS is defined as

$$\mathrm{FSS} = 1 - \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(\overline{p_{ij}} - \overline{y_{ij}})^2}{\sum_{i=1}^{M}\sum_{j=1}^{N}\overline{p_{ij}}^2 + \sum_{i=1}^{M}\sum_{j=1}^{N}\overline{y_{ij}}^2}, \quad (1)$$

where $\overline{p_{ij}}$ is the filtered forecast probability at grid point (i, j); $\overline{y_{ij}}$ is the filtered observation at grid point (i, j); and M and N are the number of rows and columns in the grid, respectively. For a batch of several examples, the FSS is defined as the average of Eq. (1) over all examples. Filtering is accomplished by taking the average over a window of 9×9 grid points, corresponding to a neighborhood width of 4 grid points (0.05° or ~5 km). Thus, the U-net is punished only when there is a mismatch of more than ~5 km between actual and predicted convection; this threshold was identified in discussions with the Taiwan CWB. We did not experiment with other neighborhood widths for the FSS. Filtering makes the FSS spatially aware

[7] When training with the full grid, we cannot use batches of more than ~8 examples; batches this small lead to instability and overfitting.

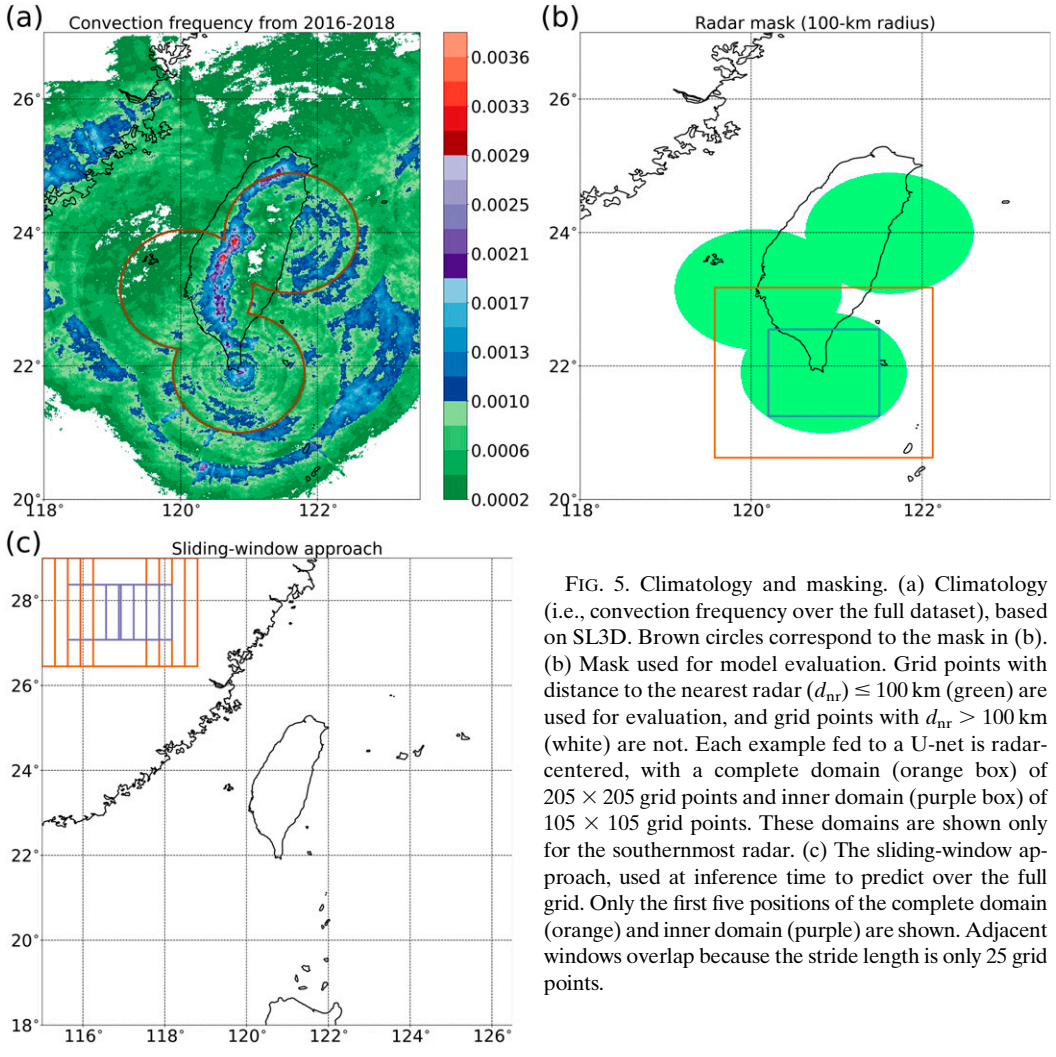

FIG. 5. Climatology and masking. (a) Climatology (i.e., convection frequency over the full dataset), based on SL3D. Brown circles correspond to the mask in (b). (b) Mask used for model evaluation. Grid points with distance to the nearest radar (d_{nr}) $\leq$ 100 km (green) are used for evaluation, and grid points with $d_{nr} > 100$ km (white) are not. Each example fed to a U-net is radar-centered, with a complete domain (orange box) of 205×205 grid points and inner domain (purple box) of 105×105 grid points. These domains are shown only for the southernmost radar. (c) The sliding-window approach, used at inference time to predict over the full grid. Only the first five positions of the complete domain (orange) and inner domain (purple) are shown. Adjacent windows overlap because the stride length is only 25 grid points.

and avoids the double-penalty problem, where the model is unduly punished for a small spatial offset between actual and forecast convection (Gilleland et al. 2009). The FSS ranges from [0, 1], and higher values are better. Although the FSS has been used widely in atmospheric science (Weusthoff et al. 2010; Sobash et al. 2011; Mittermaier et al. 2013; Bachmann et al. 2018; Ahmed et al. 2019; Loken et al. 2019; Qian and Wang 2021), it is typically used to evaluate the model post hoc, after training with a pointwise loss function. This establishes a disconnect between the model evaluation during and after training—i.e., the model is trained to optimize pointwise performance but is evaluated on spatial performance. To our knowledge, only two other studies in atmospheric science use a spatially aware evaluation score, though not the FSS, directly as the loss function (Heim and Avery 2019; Stengel et al. 2020).

d. Model evaluation

To evaluate probabilistic forecasts, we use the FSS [Eq. (1)] and attributes diagram (Hsu and Murphy 1986). Although the FSS is spatially aware by default, the attributes diagram–which is a reliability curve with extra reference lines in the background—typically is not. Thus, we redefine the reliability curve, using a neighborhood radius r_n to match forecast convection to actual convection.[8] A classic reliability curve plots forecast probability versus conditional event frequency—in this case, the frequency of actual convection given each forecast probability—and for a gridded problem the matching is pixelwise. In our reliability curves, we match each forecast convective grid point with actual convective grid points over a radius of r_n. Thus, for each forecast probability, the conditional event frequency is defined as the fraction of cases where at least one actual convective grid point occurs within r_n of the forecast. We compute the Brier score (BS) and Brier skill score (BSS) from

[8] For all scores defined in this section, if the central grid point P is within neighborhood radius r_n of a masked grid point (white in Fig. 5b), point P is ignored and not used to compute the score.

TABLE 3. Evaluation scores for binary forecasts: r_n is the neighborhood radius, a_A is the number of actual-oriented true positives (actual convective grid points for which there is forecast convection within r_n), a_F is the number of forecast-oriented true positives (forecast convective grid points for which there is actual convection within r_n), b is the number of false positives (forecast convective grid points for which there is no actual convection within r_n), and c is the number of false negatives (actual convective grid points for which there is no forecast convection within r_n).

Score	Definition	Range	Optimal value
Probability of detection (POD)	$\frac{a_A}{a_A + c}$	[0, 1]	1
False-alarm ratio (FAR)	$\frac{b}{a_F + b}$	[0, 1]	0
Frequency bias	$\frac{\text{POD}}{1 - \text{FAR}}$	[0, ∞)	1
Critical success index (CSI)	$\text{POD}^{-1} + (1 - \text{FAR})^{-1} - 1$	[0, 1]	1

the reliability curve, as in Hsu and Murphy (1986), so both scores incorporate the neighborhood radius and are spatially aware.

As mentioned in section 3c, the FSS ranges from [0, 1] and higher is better. The BS, which is the mean squared error for binary classification, ranges from [0, 1], and lower is better. The BSS, which compares the actual and climatological BS,[9] is defined as $(\text{BS}_{\text{climo}} - \text{BS})/\text{BS}_{\text{climo}}$. Like any true skill score, the BSS ranges from $(-\infty, 1]$; higher values are better; and positive values signal an improvement over climatology.

To evaluate binary forecasts, we use the contingency table. To convert probabilistic forecasts to binary, we use a probability threshold, which is not necessarily 0.5. The four scores are probability of detection (POD), false-alarm ratio (FAR), frequency bias, and critical success index (CSI). The contingency tables in this work do not include correct nulls, because we match only convective grid points (predicted to actual and vice versa), so correct nulls are ill-defined. This is similar to the setting in which the National Weather Service (NWS) evaluates tornado warnings: each case is a segment of a tornado track, and there is no such thing as a nontornado track, so there are no correct nulls. From Brooks (2004), the NWS defines the four scores as in Table 3. The variables used in Table 3 are defined schematically in Fig. 6.

e. Persistence baseline

We compare each U-net to the persistence model with the same lead time. Previous studies on convection-forecasting have typically used persistence or extrapolation as a baseline. We do not use extrapolation, because this requires complicated tracking algorithms such as optical flow or atmospheric motion vectors, which are computationally expensive and error-prone on gridded data (e.g., Héas et al. 2007). All persistence models assume that the convection mask will remain the same forever, regardless of lead time. However, we have found that Gaussian-smoothing the convection mask, with an e-folding radius of four grid points, leads to the best performance for persistence models. While the original convection mask contains only 0s and 1s, the smoothed mask contains values ranging continuously from [0, 1], which are treated as a map of forecast probabilities.

5. Model evaluation

Results of the hyperparameter experiments, used to choose the best U-net at each lead time, are shown in the supplemental material. Results in this section are for the selected U-nets only, based on testing data (year 2018), and like the loss function (section 4c), scores are computed with a four-grid-point (0.05°) neighborhood distance.

The attributes diagram (e.g., Fig. 7a)—invented by Hsu and Murphy (1986)—shows a reliability curve, inset histogram of forecast probabilities, and reference lines. The reliability curve plots conditional event frequency versus forecast probability; it answers the question: "For each forecast probability, how likely is convection to actually occur?" To create the reliability curve, we split the testing data into 20 bins: those with forecast probabilities of 0.00–0.05, 0.05–0.10, etc. A perfect reliability curve follows the diagonal gray line, where conditional event frequency equals forecast probability. Meanwhile, the vertical gray line is the climatology line; the horizontal gray line is the no-resolution line; and the blue shading is the positive-skill area, where BSS > 0, signaling an improvement over climatology. The performance diagram (e.g., Fig. 7b)—invented by Roebber (2009)—plots POD versus FAR, with each point corresponding to one probability threshold. Because frequency bias and CSI are both functions of POD and FAR (Table 3), they can be overlaid. Curves closer to the top right, where CSI is greater and frequency bias is closer to 1.0, are better.

At each lead time, to determine the best probability threshold for binary forecasts, we use the performance diagram on validation data. We start by finding two thresholds: that yielding the frequency bias closest to 1.0 (p^*_{FB}) and that yielding the highest CSI (p^*_{CSI}). If the CSI at p^*_{FB} is at least 90% of the CSI at p^*_{CSI}, we choose p^*_{FB}; otherwise, we

[9] The BS that would be achieved by a climatological model, where forecast probability is always the event frequency in the training data. Climatological frequency increases with r_n, because it is the fraction of grid points P for which there is convection within r_n of P.

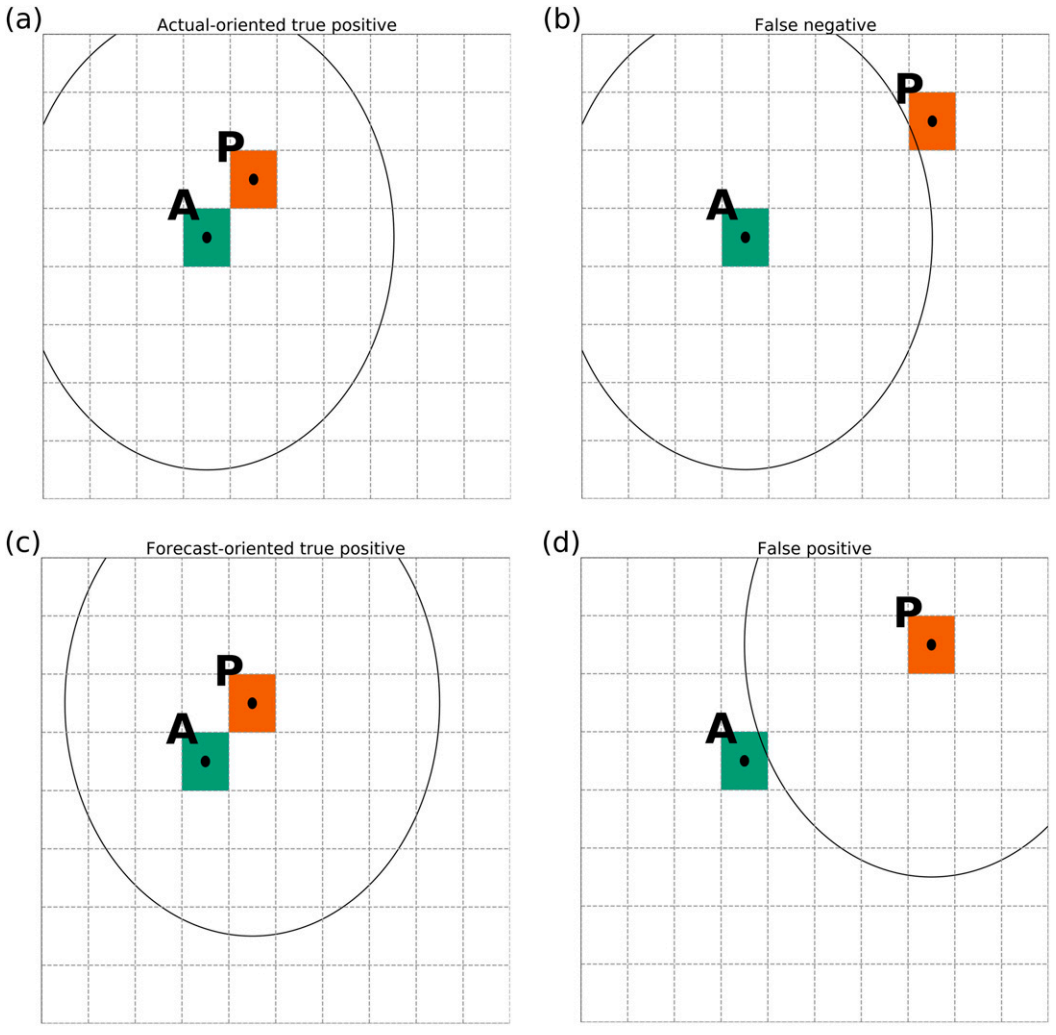

FIG. 6. Model evaluation with a neighborhood radius of four grid points. In each panel, the green (orange) box is a grid point with actual (predicted) convection. (a),(b) The actual convective grid point (A) is matched to the nearest predicted convective grid point (P). In (a), P is within four grid points of A; in (b), it is not. (c),(d) The predicted convective grid point (P) is matched to the nearest actual convective grid point (A). In (c), A is within four grid points of P; in (d), it is not.

choose p^*_{CSI}. This choice is based on the validation data, and we use the same threshold for the testing data. Thus, we treat the probability threshold as a hyperparameter.

a. 0-min lead time

Figure 7 shows domain-averaged scores for the selected U-net, which uses predictors at lag times of 0, 10, and 20 min.[10] Observations from Fig. 7 are noted here and summarized in Table 4. The attributes diagram (Fig. 7a) shows that the U-net is underconfident for probabilities $\lesssim 0.85$ and overconfident for probabilities $\gtrsim 0.85$. However, the reliability curve is almost entirely inside the positive-skill area, meaning that the U-net is more skillful than a climatological model at almost all probabilities. Also, the distance from the perfect line (i.e., difference between forecast probability and conditional event frequency) is <0.2 everywhere. In our experience this is impressive reliability for a rare event (e.g., Fig. 5 of Gagne et al. 2015, Fig. 9 of Gagne et al. 2017, Fig. 10f of Lagerquist et al. 2017, Fig. 6 of Burke et al. 2020), especially at the higher probabilities, which are rarely forecast (see inset histogram). The performance diagram (Fig. 7b) shows that the best probability threshold is 0.2, yielding a CSI of 0.375 and frequency bias of 0.893. In the monthly performance diagrams (Fig. 7c), the U-net performs best in the spring (March–May) and worst in December–January. Based on visual inspection of individual cases (not shown), we conclude that poor performance in December–January is due to a low POD, caused by the U-net missing marginal

[10] Note that the task at 0-min lead time is detection, not prediction. Nonetheless, for the sake of convenience, we use the term "prediction" throughout to describe model estimates at zero and nonzero lead times.

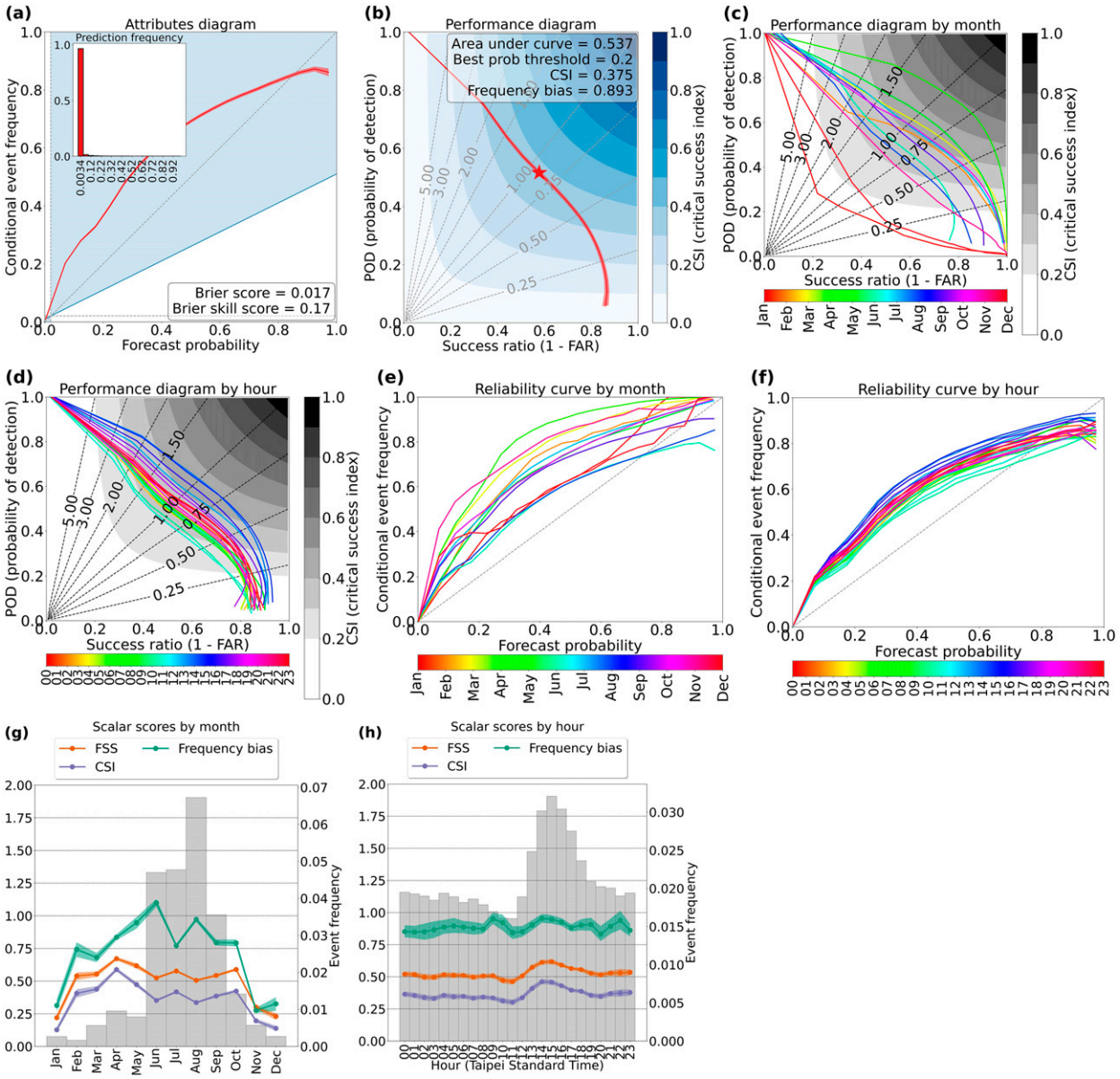

FIG. 7. Domain-averaged scores for selected 0-min U-net. (a),(b),(g),(h) The line shows the mean, and the shaded area shows the 95% confidence interval, determined by bootstrapping with 1000 replicates. (a) Attributes diagram. (b) Performance diagram. The dashed gray lines show frequency bias, and the blue color fill shows CSI. The star shows the best probability threshold (0.2), determined by considering both frequency bias and CSI. Performance diagrams split by (c) month and (d) hour. The hour is in Taipei standard time, which is 8 h ahead of UTC. Reliability curves split by (e) month and (f) hour. Other scores split by (g) month and (h) hour. For scores based on binary forecasts (frequency bias and CSI), we use a probability threshold of 0.2, corresponding to the star in (b).

convection—i.e., weak storms that barely meet the SL3D criteria for convection. Also based on visual inspection, we believe that good performance in the spring is due to thunderstorms being more discrete; discrete storms are less often obscured from the satellite by high cirrus and anvil clouds from neighboring storms. In the hourly performance diagrams (Fig. 7d), the U-net performs best from 1300 to 1759 Taipei standard time (TST) and worst from 1000 to 1159 TST. In other words, the U-net performs better when convection is more frequent (see histogram in Fig. 7h). This is unsurprising, as the performance diagram is generally worse for rare events (explained in section 5 of Lagerquist et al. 2020).

In the monthly reliability curves[11] (Fig. 7e), the U-net performs best in June, August, and December–January; in other months it is underconfident at nearly all probabilities. In the hourly

[11] The full attributes diagram is not shown, because the reference lines, which depend on climatology, would be different for every month.

TABLE 4. Summary of domain-averaged scores for selected U-net at each lead time. "AD" is attributes diagram, "PD" is performance diagram, "RC" is reliability curve, and "SS" is scalar scores.

Lead time	Evaluation method	Summary
0 min	AD (Fig. 7a)	The bad: underconfident for probabilities ≲0.85 and overconfident for probabilities ≳0.85; the good: reliability curve inside positive-skill area, within 0.2 of perfect line
	PD (Fig. 7b)	Best probability threshold is 0.2, giving CSI of 0.375 and frequency bias of 0.893
	PD by time (Figs. 7c,d)	Performs best in the spring (Mar–May) and afternoon (1300–1759 TST), worst in the winter (Dec–Jan) and late morning (1000–1159 TST)
	RC by time (Figs. 7e,f)	Performs best in the summer (Jun, Aug), winter (Dec–Jan), and late morning (1000–1159 TST); worst in the afternoon (1300–1759 TST); opposite of trends in PD by time
	SS by time (Figs. 7g,h)	Performs best in the extended summer (Apr–Oct) and afternoon, worst in the extended winter (Nov–Mar) and late morning (1000–1159 TST); similar to PD by time
30 min	AD (Fig. 9a)	Similar to 0-min U-net
	PD (Fig. 9b)	Best threshold = 0.15; CSI = 0.308; frequency bias = 1.01
	PD by time (Figs. 9c,d)	Similar to 0-min U-net
	RC by time (Figs. 9e,f)	Similar to 0-min U-net
	SS by time (Figs. 9g,h)	Similar to 0-min U-net, except bias is worst in the afternoon (see main text)
60 min	AD (Fig. 10a)	Reliability nearly perfect at all probabilities
	PD (Fig. 10b)	Best threshold = 0.2; CSI = 0.215; frequency bias = 1.05
	PD by time (Figs. 10c,d)	Similar to 0-min U-net
	RC by time (Figs. 10e,f)	Performs best in the extended summer (May–Oct) and afternoon (blue in Fig. 10f), worst in the winter (Nov–Jan); similar to trends in PD by time for 60-min U-net; however, for 0- and 30-min U-nets, trends in PD by time and RC by time were opposite; thus, at 60 min (and also beyond), there is no longer a trade-off between reliable probabilities and good binary forecasts
	SS by time (Figs. 10g,h)	Similar to 0-min U-net
120 min	AD (Fig. 13a)	Reliability curve inside positive-skill area, within 0.1 of perfect line
	PD (Fig. 13b)	Best threshold = 0.15; CSI = 0.156; frequency bias = 1.43
	PD by time (Figs. 13c,d)	Performs best in the extended summer (Apr–Oct) except May and afternoon (1300–1759 TST), worst in the extended winter (Nov–Mar) and at all times of day except afternoon; similar to 0-min U-net, except more months and hours with very poor performance
	RC by time (Figs. 13e,f)	Performs best in the summer (Jun–Sep) and late afternoon (1500–1859 TST), worst in the extended winter (Nov–Apr) and at all times of day except late afternoon; similar to 60-min U-net, except more months and hours with very poor performance
	SS by time (Figs. 13g,h)	Similar to 0-min U-net, except notably high frequency bias (see main text)

reliability curves (Fig. 7f), the U-net performs best from 1000 to 1159 TST and worst from 1300 to 1759 TST. In general, based on Figs. 7c–f, reliability is best when the performance diagram is worst, illustrating a trade-off between the quality of probabilistic and binary predictions. As noted in Table 4, this trade-off exists only for the 0- and 30-min lead times, not for longer lead times. Figures 7g–h show scalar scores by month and hour, which are generally best from April to October and in the afternoon, worst from November to March and in the late morning. In other words, the three scores support the conclusion from the performance diagrams, that performance is better when convection is more frequent. However, two of the three scores (CSI and frequency bias) are redundant with the performance diagram.

Figure 8 shows gridded scores for the selected U-net. In Figs. 8a,f the optimal value is 1.0 and the scores nearest 1.0 generally occur along the west side of the mountains, where convection is frequent due to orographic lifting. Figures 8a–f show many arc-shaped artifacts due to radar coverage (especially around the northernmost radar), as well as Fig. 8g, which shows the label-based climatology (convection frequency according to SL3D). However, in the model-based climatology (mean convection probability from the U-net; Fig. 8h), these artifacts are absent. According to the U-net, the climatological maxima occur along the west side of the mountains and over warm sea surface temperatures (SST) near the south of Taiwan, while the minima occur over cool SSTs near the north of Taiwan.[12] Thus, we consider the model-based climatology more plausible than the label-based climatology. This is an advantage of the U-net over SL3D, although the advantage is subjective and not quantifiable in an evaluation score.

b. 30-min lead time

Figure 9 shows domain-averaged scores for the selected U-net, which uses predictors at lag times of 0, 20, 40, and 60 min. Table 4 summarizes differences between domain-averaged scores for the 0- and 30-min U-nets. One difference is that frequency bias for the 30-min U-net is worst in the afternoon (Fig. 9h), when FSS and CSI are best. This problem, as well as poor frequency bias in the winter months (Fig. 9g), could be alleviated by choosing a different probability threshold for each month/hour. For research purposes

[12] See Fig. 2 of Sun et al. (2019) for a climatology of SST in the area.

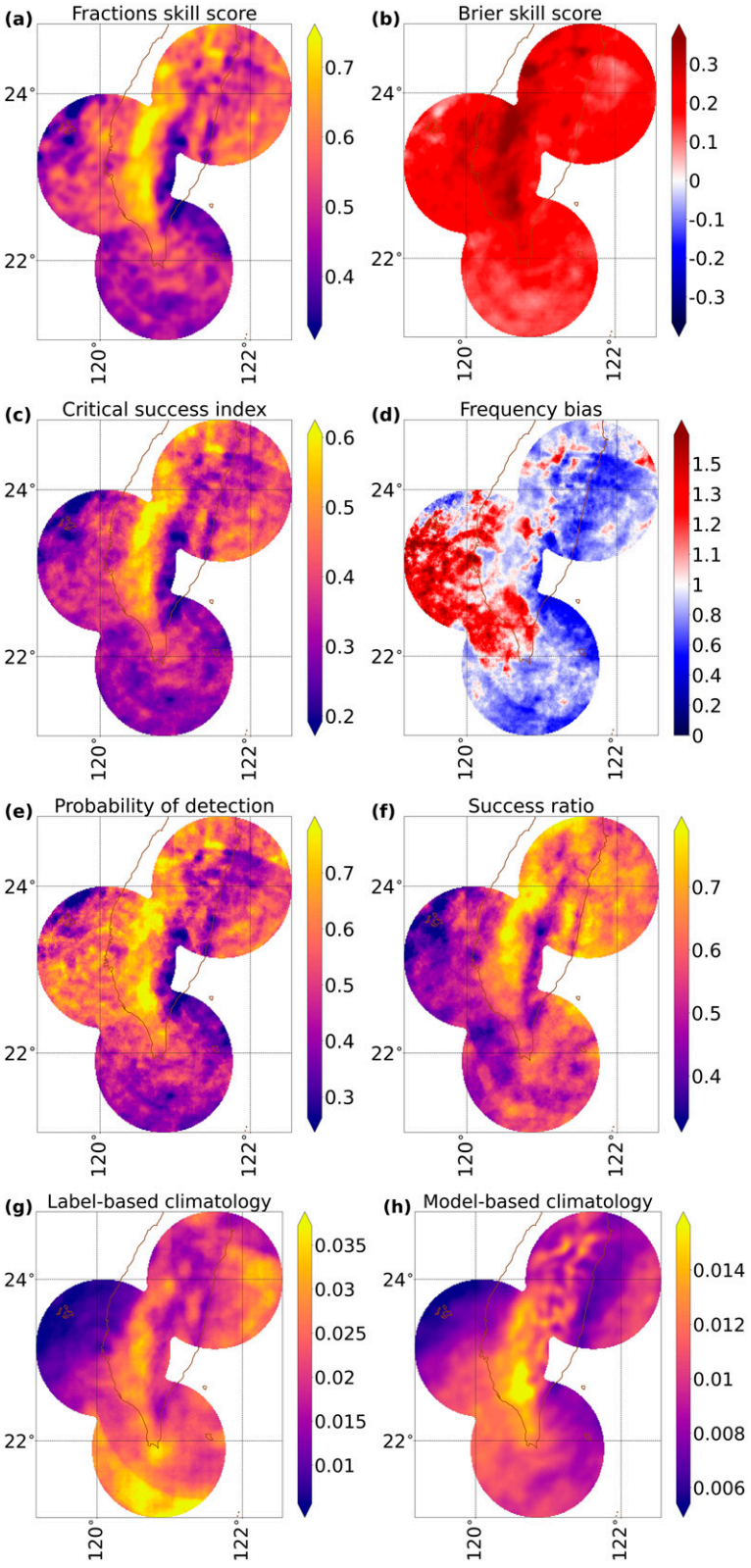

FIG. 8. Gridded scores for selected 0-min U-net. (c)–(f) For scores based on binary forecasts, we use a probability threshold of 0.2, corresponding to the star in Fig. 7b.

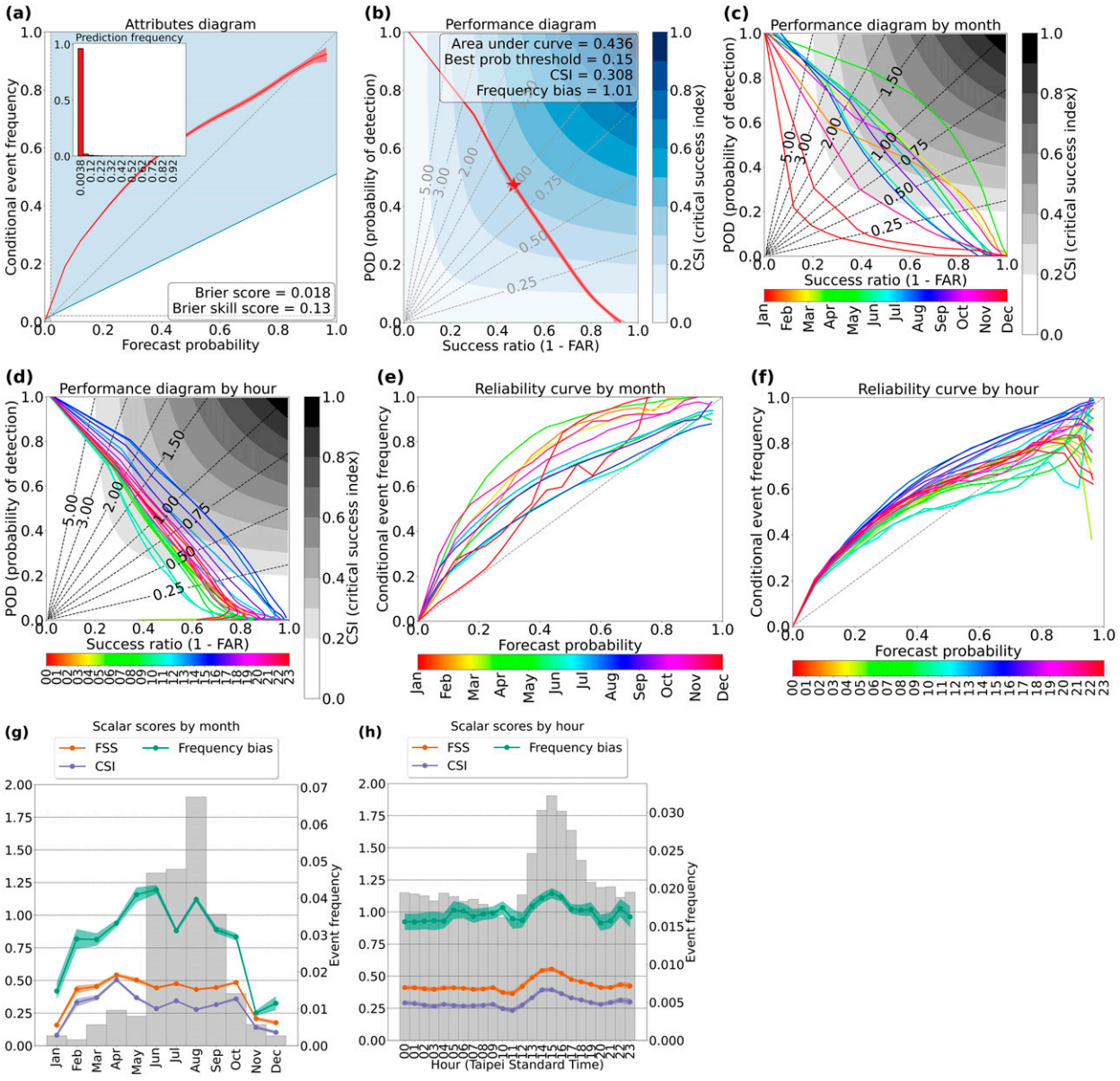

FIG. 9. Domain-averaged scores for selected 30-min U-net. Formatting is explained in the caption of Fig. 7. (g),(h) For scores based on binary forecasts (frequency bias and CSI), we use a probability threshold of 0.15, corresponding to the star in (b).

this would add a lot of hyperparameters, so we have chosen to keep one probability threshold per model. However, in an operational setting where binary forecasts are required, varying the probability threshold by month/hour would be useful. In such a setting we would also consider the disparate cost of false positives versus negatives (i.e., false negatives are more costly, as failing to take preventive action for a thunderstorm hazard could be fatal) in choosing the probability threshold.

Differences between the 30-min U-net and persistence model are discussed in section c of the supplemental material. The main conclusion is that, although the persistence model outperforms the U-net on all objective scores, the U-net produces a more plausible spatial climatology (i.e., gridded map of convection frequency).

c. 60-min lead time

Figure 10 shows domain-averaged scores for the selected U-net, which uses predictors at lag times of 0, 20, and 40 min. Table 4 summarizes differences between domain-averaged scores for the 0- and 60-min U-nets; observations not included in Table 4 are noted here. Reliability curves for November–January are truncated (Fig. 10e), because the U-net never forecasts probabilities >0.5 in these months. In the hourly reliability curves (Fig. 10f), outside of the afternoon (blue), the U-net is generally unreliable (either

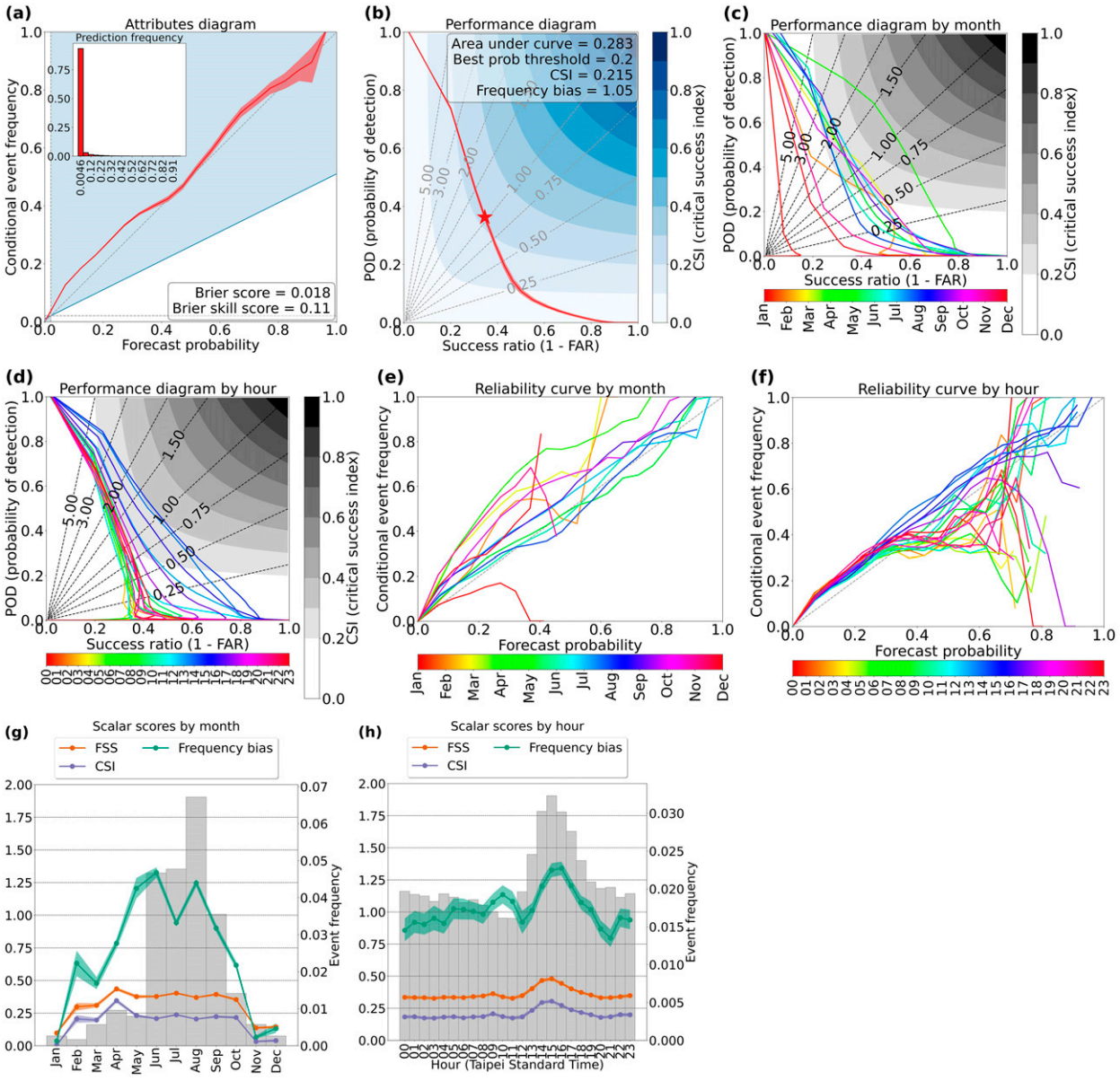

FIG. 10. Domain-averaged scores for selected 60-min U-net. Formatting is explained in the caption of Fig. 7. (g),(h) For scores based on binary forecasts (frequency bias and CSI), we use a probability threshold of 0.2, corresponding to the star in (b).

overconfident or underconfident) when forecasting probabilities $\gtrsim 0.4$. Thus, the overall reliability curve (Fig. 10a) does not tell the complete story, obscuring problems that occur for certain months and hours, generally those with the least convection.

Figure 11 shows domain-averaged scores for the persistence model, to be compared with Fig. 10 for the U-net. In the attributes diagram (Fig. 11a), the persistence model clearly has a worse BS and BSS, worse underconfidence at low probabilities, and worse overconfidence at high probabilities. However, the persistence model has a slightly better performance diagram (Fig. 11b), with a CSI 0.021 higher. In the monthly and hourly performance diagrams (Figs. 11c,d), again the two models are similar; the persistence model is better only for the worst months, November–January. In the monthly and hourly reliability curves (Figs. 11e,f), the persistence model has worse underconfidence at low probabilities and worse overconfidence at high probabilities, similar to the overall attributes diagram. In the monthly and hourly plots of scalar scores (Figs. 11g,h), the U-net generally has a better FSS; the persistence model generally has a better CSI; and the persistence model generally has a much better frequency bias, especially for the worst months and hours.

Meanwhile, Fig. 12 shows gridded scores for the two 60-min models. Evaluation scores (Figs. 12a–f) for the two 60-min models are close, but as for the 30-min models (discussed in section c of the supplemental material), the U-net-based

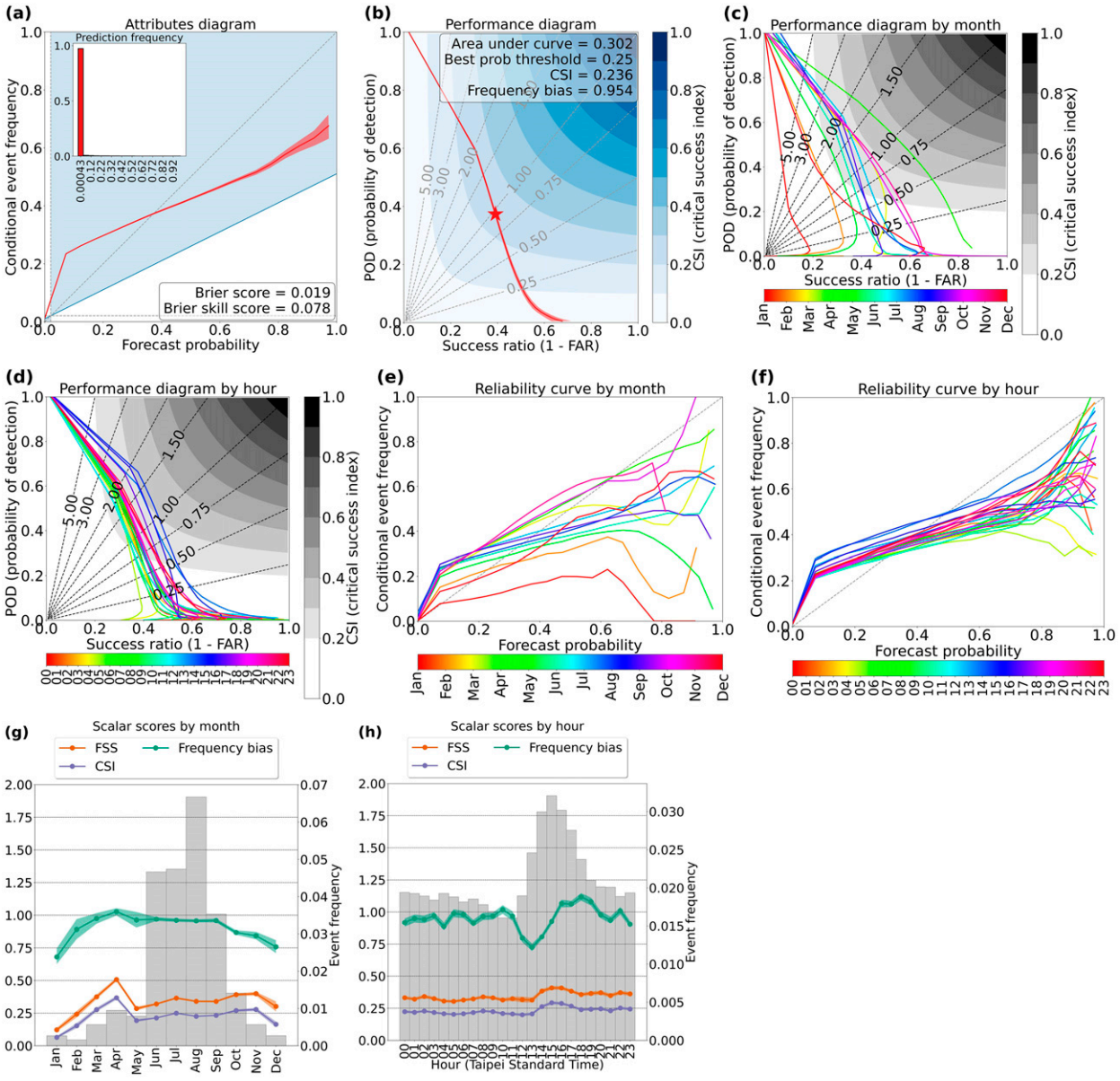

FIG. 11. Domain-averaged scores for 60-min persistence model. Formatting is explained in the caption of Fig. 7. (g),(h) For scores based on binary forecasts (frequency bias and CSI), we use a probability threshold of 0.25, corresponding to the star in (b).

climatology is more plausible than the persistence-based climatology (Figs. 12g,h). Specifically, the persistence-based climatology contains many radar artifacts, because the persistence model mimics the convection mask derived from radar data. This subjective advantage is difficult to quantify when the labels used for evaluation (Fig. 8g) are imperfect.

Overall, based on much better reliability and a more plausible climatology, we conclude that the 60-min U-net is better than the 60-min persistence model.

d. Longer lead times

Results for the 90-min models (U-net and persistence) are relegated to supplemental Figs. S14–S16, because they are very similar to those for the 60-min models. The two main differences are (i) the 90-min models perform worse than the 60-min models, although the 90-min U-net still has impressive overall reliability (supplemental Fig. S14a); (ii) the advantage of the U-net over the persistence model is greater at 90 min than at 60 min.

Figure 13 shows domain-averaged scores for the selected 120-min U-net, which uses predictors at lag times of 0 and 20 min. Table 4 summarizes differences between domain-averaged scores for the 0- and 120-min U-nets. One difference is that in the monthly and hourly plots of scalar scores (Figs. 13g,h), frequency bias is notably high (near 2.0) when FSS and CSI are best, unlike for other lead times. This is because the chosen probability threshold, based on

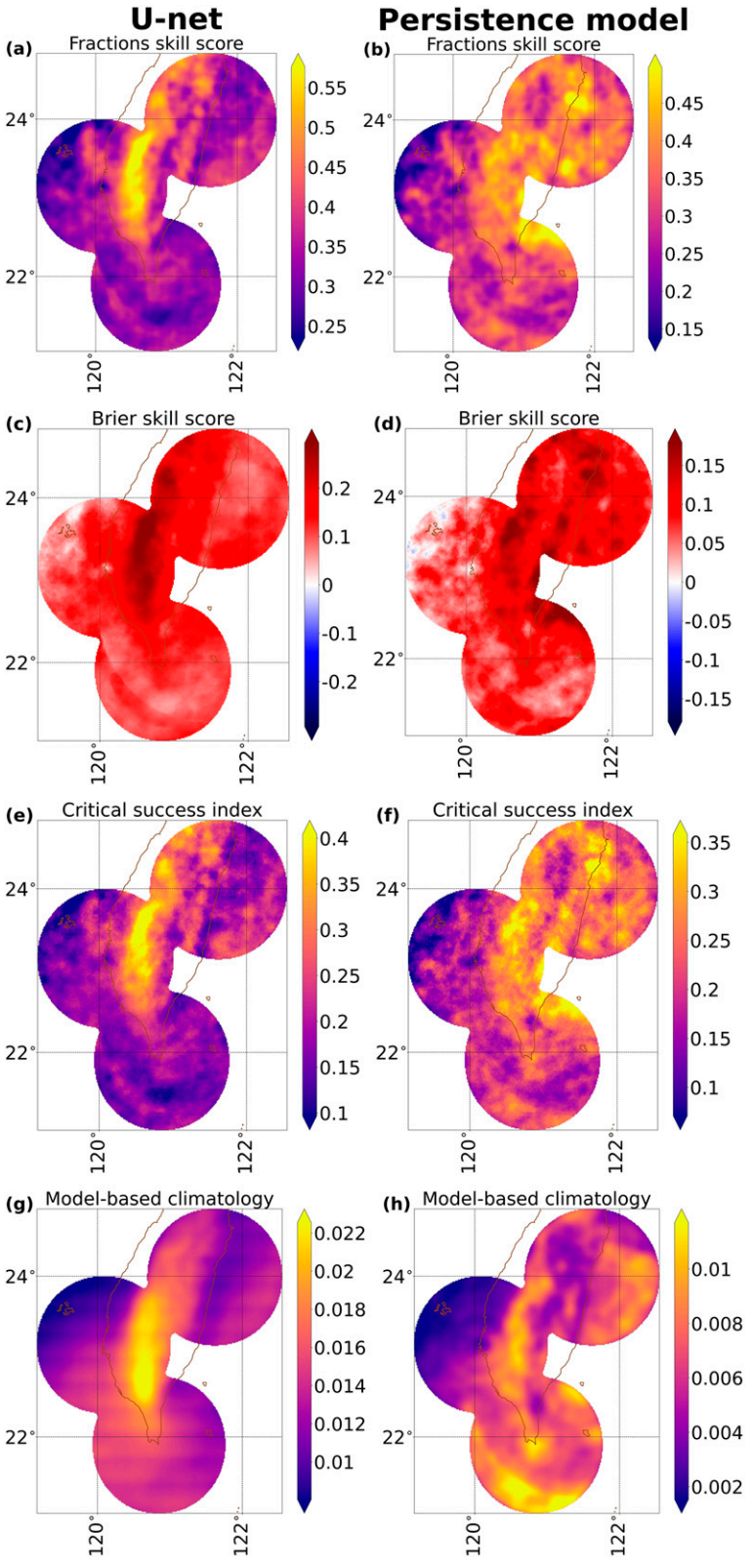

FIG. 12. Gridded scores for selected 60-min U-net and 60-min persistence model.

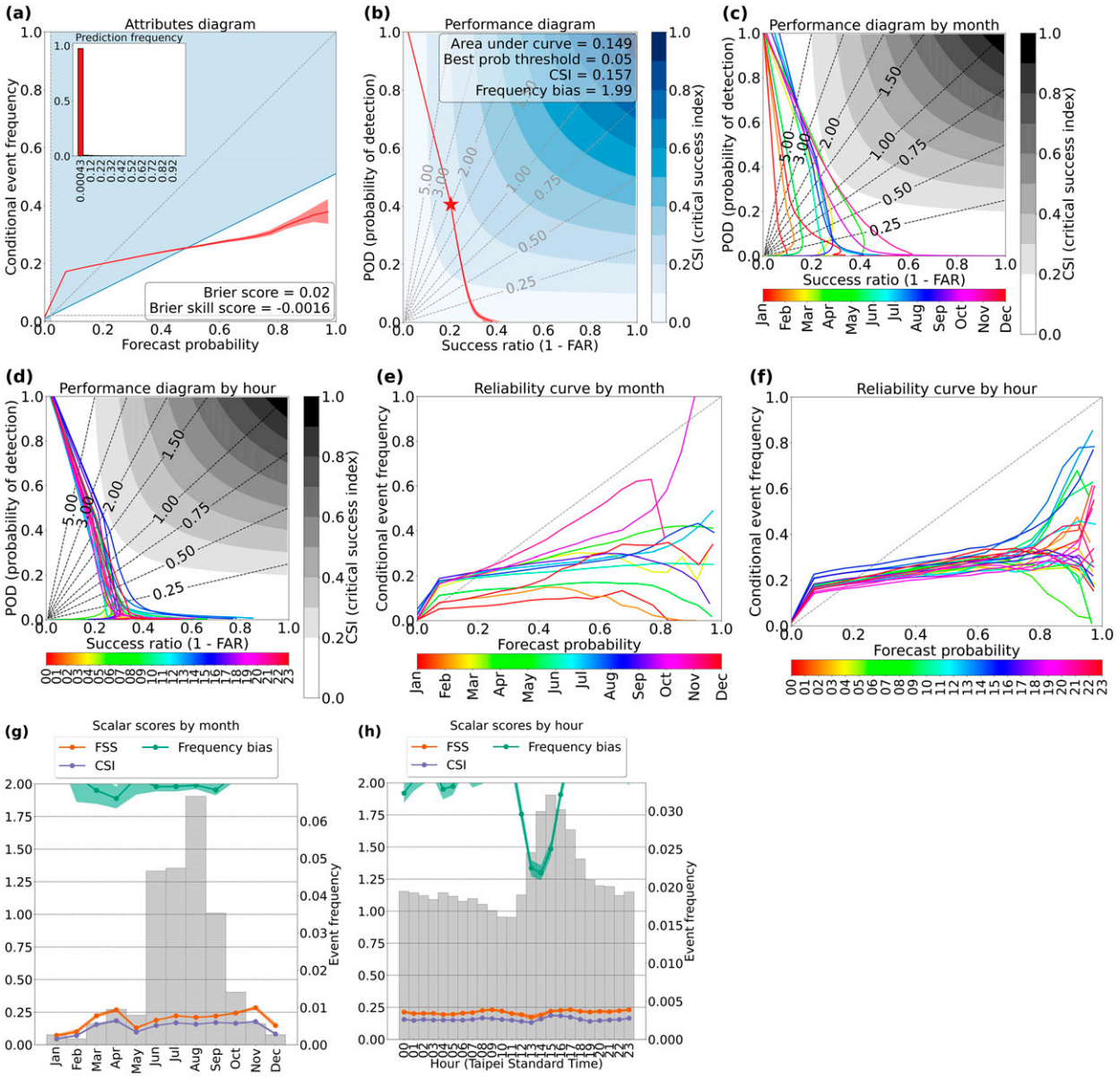

FIG. 13. Domain-averaged scores for selected 120-min U-net. Formatting is explained in the caption of Fig. 7. (g),(h) For scores based on binary forecasts (frequency bias and CSI), we use a probability threshold of 0.15, corresponding to the star in (b).

the validation data, is too low for the testing data, yielding an overall frequency bias of 1.43 (shown in Fig. 13b).

In supplemental Fig. S17 the domain-averaged scores for the persistence model are shown, to be compared with Fig. 13 for the U-net. The U-net clearly outperforms the persistence model in all facets, especially reliability (Figs. 13a,e,f). Meanwhile, supplemental Fig. S18 shows gridded scores for the two 120-min models. Overall conclusions, especially the U-net-based climatology being more plausible, are the same as for other lead times.

e. *Summary of all lead times*

Figure 14 summarizes the performance of the U-nets and persistence models versus lead time. Consistent with the foregoing discussion, Fig. 14 shows that, in terms of all scores except CSI, the U-net overtakes the persistence model at 60-min lead time and more dramatically outperforms persistence at lead times >60 min.

f. *Case studies*

Figures 15–17 show three case studies, each in a different month of the testing year (2018). Figure 15 shows a winter case, where there is only one actual thunderstorm, labeled "A." The storm is weak, with a small reflectivity core and reflectivity barely exceeding 45 dBZ (Fig. 15b). The 0-min U-net (Fig. 15a) estimates probabilities of 0.15–0.20 around storm A and probabilities < 0.05 everywhere else, indicating good discrimination. U-nets at nonzero lead times (left column)

forecast probabilities <0.05 everywhere, so do not provide good discrimination. The persistence models (right column) also forecast probabilities <0.05 everywhere, except the 30- and 90-min persistence models forecast high probabilities near storm A. Locations where the 30- and 90-min persistence models forecast nonzero probabilities, are locations where storms existed 30 and 90 min before the valid time, respectively. By visual inspection, we have found that both of these storms are earlier snapshots of storm A. Also by visual inspection, storm A exists at 60 and 120 min before the valid time; however, it does not meet the SL3D criteria and is therefore labeled nonconvective, so the 60- and 120-min persistence models forecast zero everywhere. Such imperfections in the SL3D labels affect both the U-nets and persistence models, but they affect the persistence models more strongly. A persistence model initialized at t_0 must use the SL3D labels at t_0, while a U-net draws from its experience over the entire training set and therefore can partly overcome incorrect labels. In general, because winter thunderstorms are marginal ("barely convective"), they are difficult for both the U-nets and persistence models to forecast.

Figure 16 shows a June case, with four thunderstorms around the southern radar (Fig. 16b). The 0-min U-net (Fig. 16a) estimates probabilities >0.75 around storm A, >0.5 around storm B, up to 0.5 around storm C, and up to 0.25 around storm D. Almost everywhere else, the 0-min U-net has probabilities <0.05. Thus, as in Fig. 15, the 0-min U-net has good discrimination. At nonzero lead times (Figs. 16c–j), there are two main differences between the U-nets and persistence models. First, the U-nets generally make better predictions for storms C and D, especially at 30-min lead time, where the persistence model has only zeros around C and D. This is a case of convective initiation, which the U-net forecasts with probabilities up to 0.2 around storm C and 0.1 around storm D. At lead times beyond 30 min, the U-nets also outperform the persistence models in this area, with the persistence models missing the initiation of D and missing the location of C. Second, both the U-nets and persistence models produce false alarms southwest of Taiwan, but the U-nets' false alarms cover a larger area.

Figure 17 shows an August case, during the passage of Tropical Depression Luis. The strongest convection (according to composite reflectivity; Fig. 17b) occurs in two areas, labeled A and B. The 0-min U-net (Fig. 17a) estimates high probabilities in both areas, but some high probabilities are false alarms, like those in area C. However, note that composite reflectivity > 35 dBZ (often used as the definition of convection; section 1) in most of area C. Thus, the discrepancy between the 0-min U-net and SL3D labels here is probably related to the subjectivity of defining convection, as well as the difficulty of making a radar-based and satellite-based definition agree. The 0-min U-net also has a large area of false negatives on the east side of Luis, where convection is weaker and appears to be embedded in an area of stratiform rain. As the lead time increases, U-net probabilities (left column) lose sharpness, with the maximum probability decreasing from ~1.0 at 0-min lead time to ~0.5 at 120-min lead time. The persistence models (right column) do not lose sharpness as lead time increases, but their high probabilities are generally misplaced. This is especially true at 120 min, where the U-net probabilities are highest in areas A and B (Fig. 17i), while the persistence-model probabilities are highest in area D, which is mostly nonconvective. Also, the strong convection in area A is completely missed by the persistence model. This problem could potentially be alleviated by storm-tracking—i.e., extrapolating storm locations into the future, rather than assuming no movement—as the rainbands corresponding to area A exist at 120 min before the valid time. However, this would be a difficult tracking problem. Individual storm cells in the rainband mostly do not last for 120 min, and nearly all tracking algorithms focus on individual cells. A multiscale tracking algorithm would be needed, capable of tracking features such as storm cells, rainbands, and possibly entire cyclones. Reasoning with features at multiple scales is already a strength of the U-net (section 3a), which is likely why it so dramatically outperforms the persistence model at 120-min lead time.

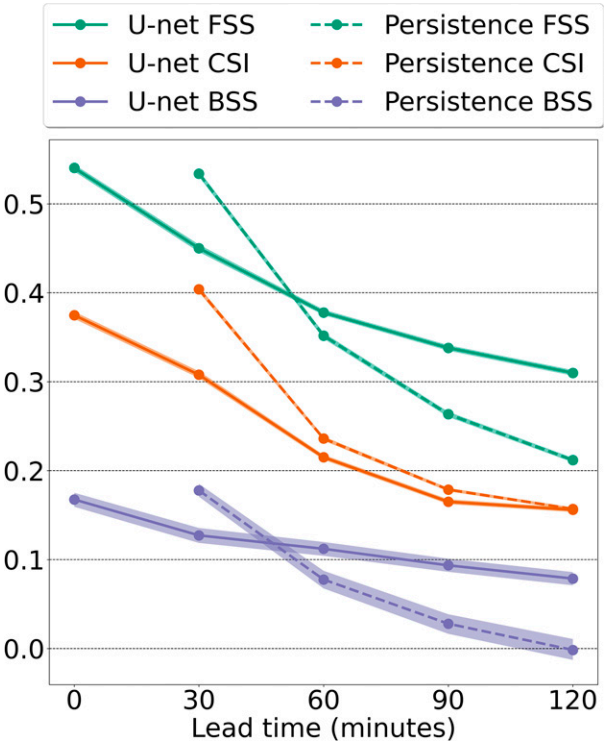

FIG. 14. Summarized performance of U-nets and persistence models vs lead time. For each score, the line shows the mean, and the shaded area shows the 95% confidence interval, determined by bootstrapping with 1000 replicates. We do not show a 0-min persistence model, because a 0-min persistence model would have perfect performance (BSS = FSS = CSI = 1) by definition. For each model $\mathcal{F}$, we have used the best probability threshold, previously determined on validation data, to compute CSI.

6. Model interpretation

The permutation test measures the importance of each predictor x_j by measuring how much the performance of a trained model declines when x_j is permuted—i.e., randomly shuffled across data examples, so that maps of x_j are spatiotemporally intact but assigned to the wrong examples. In this case we measure the importance of each *Himawari-8* spectral band, averaged

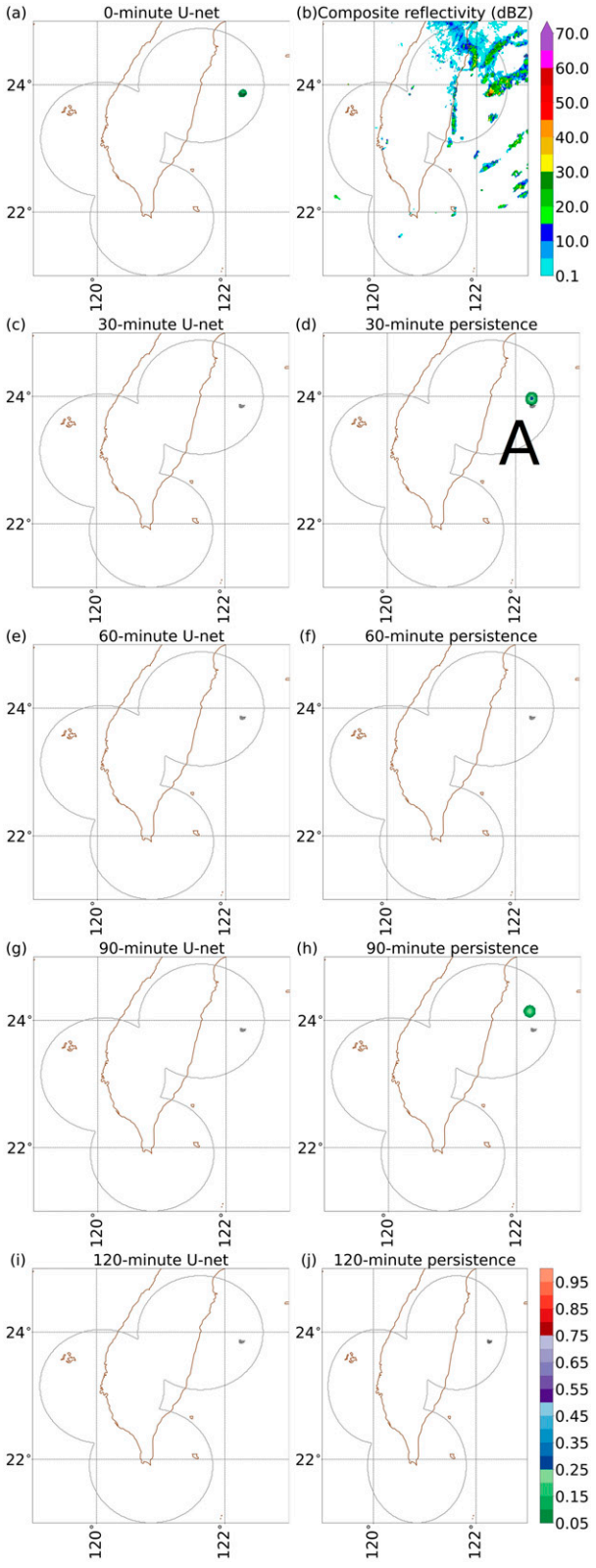

FIG. 15. Convection probabilities valid at 2230 UTC 25 Jan 2018 (0630 TST 26 Jan 2018). In all panels, black dots show actual convection at 2230 UTC, according to SL3D. Black dots are not shown outside the 100-km range rings (gray circles), because SL3D labels here are ignored. The letter label ("A") is explained in the main text. (a) Estimated probabilities from 0-min U-net. (b) Composite reflectivity at 2230 UTC. (c)–(j) Forecast probabilities at nonzero lead times from both U-nets and persistence models. All forecasts are valid at the same time (2230 UTC), so an *N*-min forecast was initialized at *N* min before 2230 UTC.

over the other dimensions, which are lag times and grid points. We run the permutation test on 30 randomly selected days from the testing set, using the same loss function as during training (FSS with a window of 9×9 grid points). We run all four versions of the test—single-pass forward, multipass forward, single-pass backward, and multipass backward—for the U-net at each lead time. The four versions handle correlated predictors differently, so it is useful to run all four and look for consistent results. See McGovern et al. (2019) for details on the permutation test.

Results for the 0-min U-net are shown in Fig. 18; results for other lead times, which are similar, are shown in supplemental Figs. S19–S22. In each panel of these figures, predictor importance increases toward the top. For the 0-min U-net—ignoring the multipass backward test, for which most results are not statistically significant—band 13 is the most important for all three tests, while band 11 is second-most important for two tests and third-most important for one test. Considering all lead times, bands 11, 13, and 16 appear most often with statistical significance in the top three predictors. As shown in Fig. 1 of Da (2015), weighting functions for these bands peak in the lower troposphere, while those for bands 8–10 peak in the middle to upper troposphere. Thus, the most important information for forecasting convection is in the lower troposphere. If the task were instead *strong* convection (i.e., thunderstorms with deep updrafts), we suspect that bands 8–10 would be more important, as in Molina et al. (2021), who found that midlevel fields are more important for strong thunderstorms than for weak thunderstorms.

7. Summary and future work

We applied U-nets, a type of deep-learning model, to forecast convection around Taiwan at lead times up to 120 min. The predictors are a time series of "brightness-temperature images" from the *Himawari-8* satellite, and the labels are a binary convection mask, produced by applying an echo-classification algorithm called SL3D to radar data. We experimented with three U-net architectures: vanilla, temporal, and U-net++. We found that the vanilla architecture performs best, based on multiple scores. At each lead time (0, 30, 60, 90, and 120 min) we tuned other hyperparameters, including the lag times for predictors. The best model at each lead time uses predictors at two or more lag times, indicating that the time series is important. Also, the permutation test indicates that spectral bands weighted toward the lower troposphere are more important predictors than those weighted toward the middle to upper troposphere. Our novel contributions include 1) applying U-nets to forecast convection; 2) experimenting with novel U-net architectures

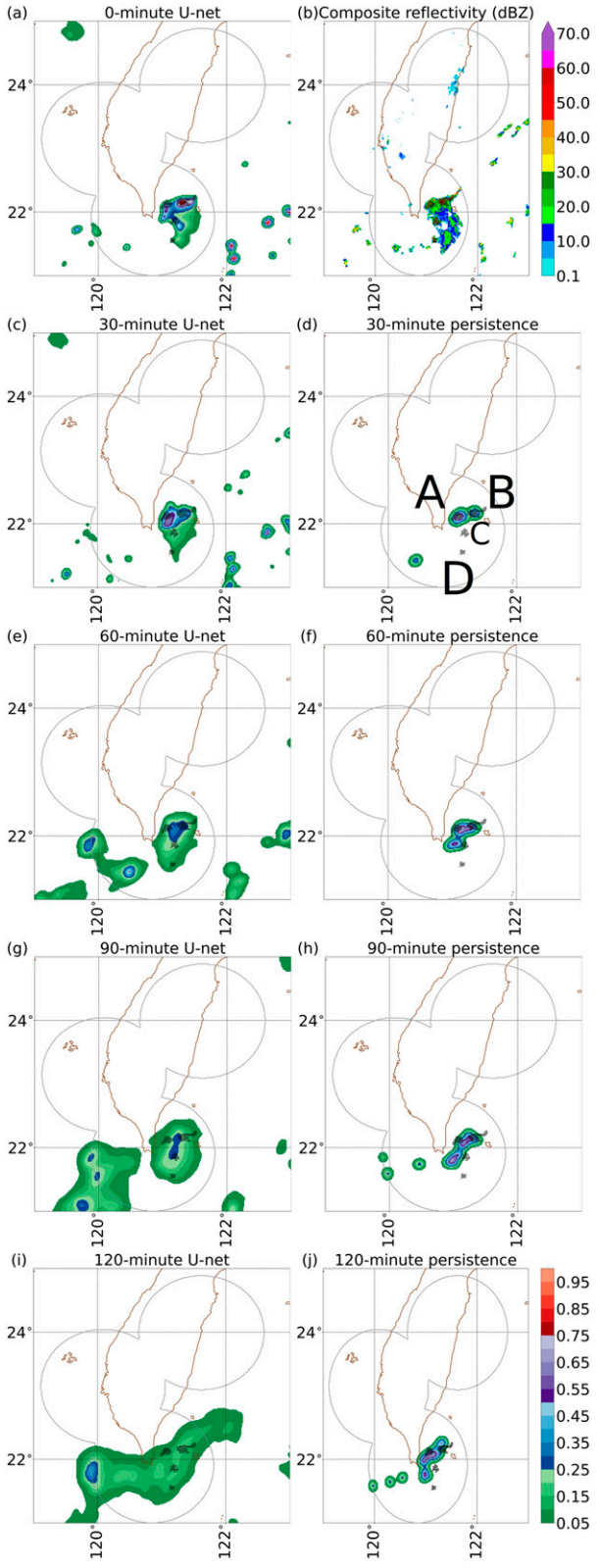

FIG. 16. Convection probabilities valid at 1920 UTC 3 Jun 2018 (0320 TST 4 Jun 2018). Letter labels (''A''–''D'') are explained in the main text. Other formatting is explained in the caption of Fig. 15.

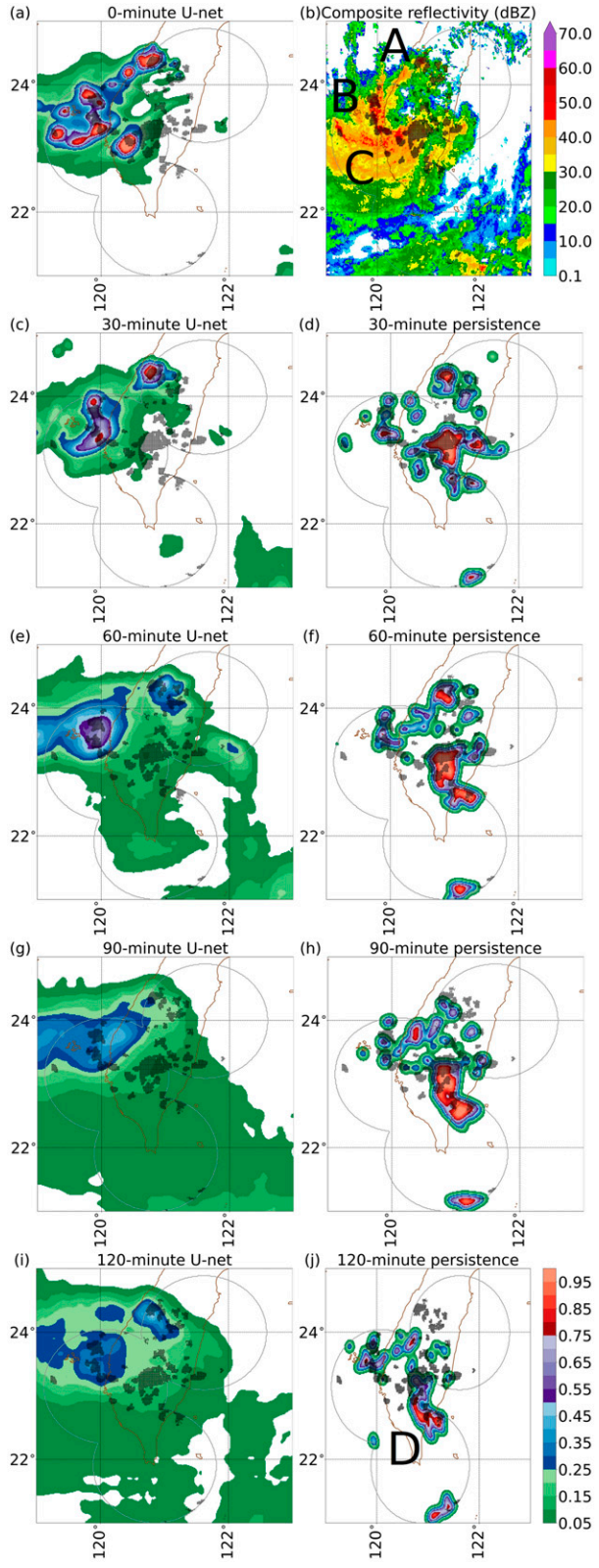

FIG. 17. Convection probabilities valid at 0830 UTC 23 Aug 2018 (1630 TST 23 Aug 2018). Letter labels (A to D) are explained in the main text. Other formatting is explained in the caption of Fig. 15.

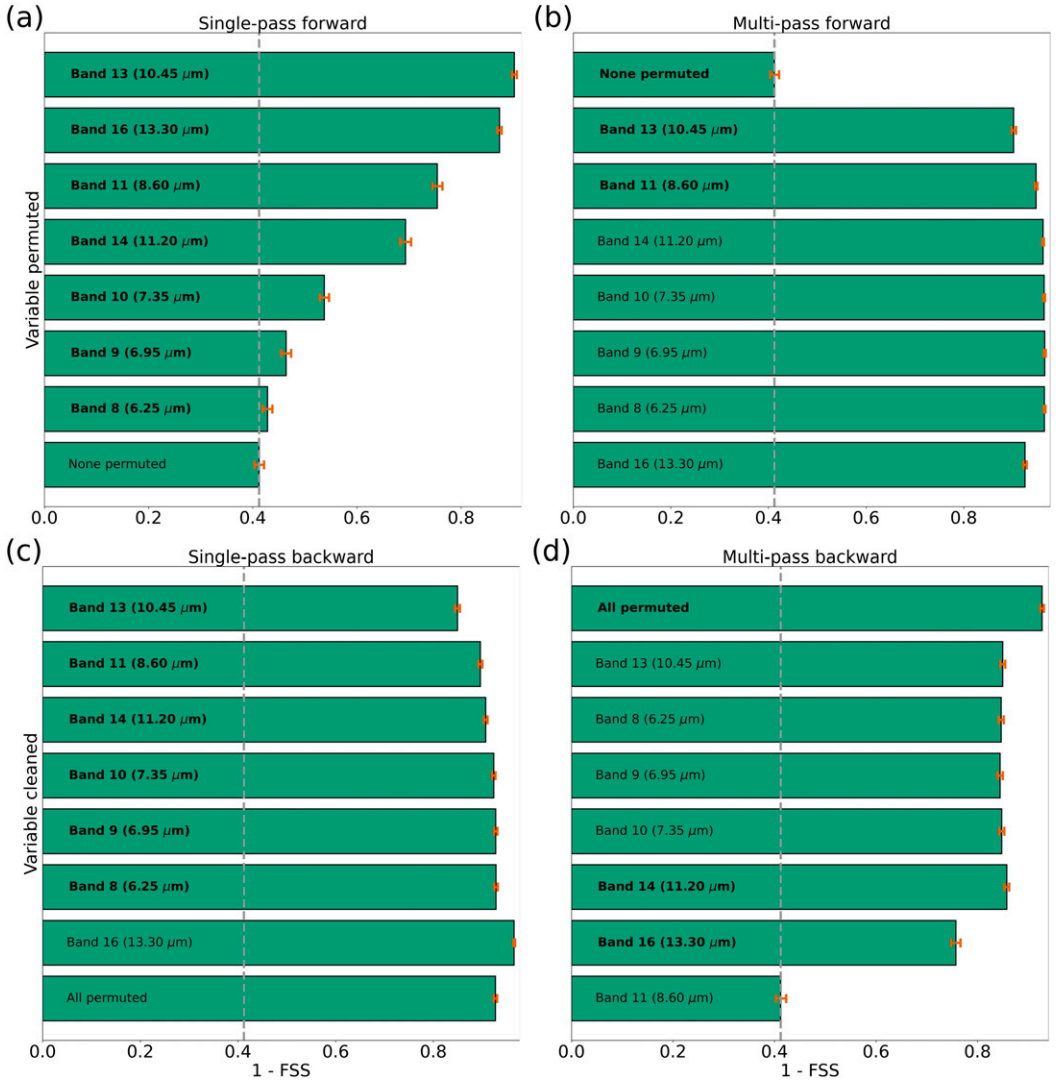

FIG. 18. Results of permutation test for 0-min U-net. The value for the bar labeled "xj" has a different meaning in each panel: (a) the loss (1 − FSS) with only predictor x_j permuted; (b) the loss with x_j and all predictors above permuted; (c) the loss with only x_j in the correct order and all other predictors permuted; (d) the loss with x_j and all predictors above in the correct order. The diagonal gray line is the original loss, with all predictors in the correct order. If x_j is in boldface font, it is significantly more important (at the 95% confidence level) than the predictor below it, based on a paired-bootstrapping test with 1000 iterations. Orange error bars show the 95% confidence interval, also based on bootstrapping with 1000 iterations.

for atmospheric science; 3) using a spatially aware evaluation score (FSS) as the loss function, not only for post hoc evaluation; 4) the sliding-window approach, wherein we train on patches of the *Himawari-8* domain with adequate radar data, then apply the trained models to the full domain.

For the 60-min lead time, we expanded the hyperparameter experiment documented in the supplemental material, training 180 vanilla U-nets with a pixelwise loss function instead of FSS. Namely, we used pixelwise cross-entropy, which is a common choice for binary classification. We found that the best U-net trained with pixelwise cross-entropy has a similar performance diagram, but much worse attributes diagram, than the best U-net trained with FSS (supplemental Fig. S23). Specifically, the best U-net trained with pixelwise cross-entropy is extremely overconfident when forecasting any probability $\gtrsim 0.3$.

For model evaluation during and after training, we used neighborhood-based scores, with a neighborhood radius of 0.05°, to avoid problems such as the double penalty. We compared the U-net at each nonzero lead time to a persistence model at the same lead time. We found that the U-net is worse than the persistence model at 30-min lead time, slightly better at 60 min, and markedly better at 60+ min. To our knowledge, two previous works have developed convection-forecasting algorithms

for the *Himawari-8* (Lee et al. 2017; Han et al. 2019). Although they achieve better evaluation scores, we cannot directly compare our results with theirs, because (i) they forecast convective initiation only; (ii) they forecast over South Korea rather than Taiwan; (iii) they use different evaluation methods, including an object-oriented, rather than gridded, approach; (iv) they use much smaller testing sets, i.e., less than 10 days.

Future work will investigate season-specific models, especially for improving the prediction of marginal convection in the winter. There are many fewer storms in the winter, which may lead to problems with sample size, so the winter model will likely be trained with examples from all seasons but using the "statistical weighting" scheme in Burke et al. (2021), where winter (summer) examples have the highest (lowest) weight. Also, future work will incorporate other predictors, such as NWP output and mesoanalysis based on in situ observations. We believe that this is the main avenue for improvement, as these data indicate how conducive the environment is to nonlinear processes such as convective initiation and decay.

Acknowledgments. We thank the Taiwan CWB for providing the satellite and radar data used herein. This work was partially supported by the NOAA Global Systems Laboratory, Cooperative Institute for Research in the Atmosphere, and NOAA Award NA19OAR4320073. Author Ebert-Uphoff's work was partially supported by NSF AI Institute Grant ICER-2019758 and NSF Grant OAC-1934668.

Data availability statement. Input data (satellite and radar images) are available upon request from the authors, as well as trained versions of the selected models (best U-net at each lead time). We used version 1.0.0 of ML4convection (doi: 10.5281/zenodo.4673642)—a Python library managed by author Lagerquist—to train, evaluate, and interpret all models (U-nets and persistence models) in this work. Since the U-net architectures are complicated, for each lead time we have included a script that creates the architecture for the best U-net. These can be found at scripts/make_best_architecture_*minutes.py in the Python library.

REFERENCES

Ahijevych, D., J. Pinto, J. Williams, and M. Steiner, 2016: Probabilistic forecasts of mesoscale convective system initiation using the random forest data mining technique. *Wea. Forecasting*, **31**, 581–599, https://doi.org/10.1175/WAF-D-15-0113.1.

Ahmed, K., D. Sachindra, S. Shahid, M. Demirel, and E. Chung, 2019: Selection of multi-model ensemble of general circulation models for the simulation of precipitation and maximum and minimum temperature based on spatial assessment metrics. *Hydrol. Earth Syst. Sci.*, **23**, 4803–4824, https://doi.org/10.5194/hess-23-4803-2019.

Bachmann, K., C. Keil, and M. Weissmann, 2018: Impact of radar data assimilation and orography on predictability of deep convection. *Quart. J. Roy. Meteor. Soc.*, **145**, 117–130, https://doi.org/10.1002/qj.3412.

Brooks, H., 2004: Tornado-warning performance in the past and future: A perspective from signal detection theory. *Bull. Amer. Meteor. Soc.*, **85**, 837–844, https://doi.org/10.1175/BAMS-85-6-837.

Burke, A., N. Snook, D. Gagne, S. McCorkle, and A. McGovern, 2020: Calibration of machine learning-based probabilistic hail predictions for operational forecasting. *Wea. Forecasting*, **35**, 149–168, https://doi.org/10.1175/WAF-D-19-0105.1.

——, ——, and A. McGovern, 2021: Improving machine learning-based probabilistic hail forecasts through statistical weighting. *Conf. on Artificial Intelligence for Environmental Science*, virtual, Amer. Meteor. Soc., J5.8, https://ams.confex.com/ams/101ANNUAL/meetingapp.cgi/Paper/383903.

Chang, P., P. Lin, B. Jou, and J. Zhang, 2009: An application of reflectivity climatology in constructing radar hybrid scans over complex terrain. *J. Atmos. Oceanic Technol.*, **26**, 1315–1327, https://doi.org/10.1175/2009JTECHA1162.1.

Chen, Y., L. Bruzzone, L. Jiang, and Q. Sun, 2021: ARU-Net: Reduction of atmospheric phase screen in SAR interferometry using attention-based deep residual U-net. *IEEE Trans. Geosci. Remote Sens.*, **59**, 5780–5793, https://doi.org/10.1109/TGRS.2020.3021765.

Chiu, H., E. Adeli, and J. Niebles, 2020: Segmenting the future. *IEEE Rob. Autom. Lett.*, **5**, 4202–4209, https://doi.org/10.1109/LRA.2020.2992184.

Da, C., 2015: Preliminary assessment of the Advanced Himawari Imager (AHI) measurement onboard Himawari-8 geostationary satellite. *Remote Sens. Lett.*, **6**, 637–646, https://doi.org/10.1080/2150704X.2015.1066522.

Fukushima, K., 1980: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.*, **36**, 193–202, https://doi.org/10.1007/BF00344251.

——, 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, J. Brotzge, M. Coniglio, J. Correia, and M. Xue, 2015: Day-ahead hail prediction integrating machine learning with storm-scale numerical weather models. *Conf. on Artificial Intelligence*, Austin, TX, Association for the Advancement of Artificial Intelligence, 7 pp., http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.696.667&rep=rep1&type=pdf.

——, ——, S. Haupt, R. Sobash, J. Williams, and M. Xue, 2017: 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.

Gilleland, E., D. Ahijevych, B. Brown, B. Casati, and E. Ebert, 2009: Intercomparison of spatial forecast verification methods. *Wea. Forecasting*, **24**, 1416–1430, https://doi.org/10.1175/2009WAF2222269.1.

Goodfellow, I., Y. Bengio, and A. Courville, 2016: *Deep Learning*. MIT Press, 773 pp., https://www.deeplearningbook.org.

Han, D., J. Lee, J. Im, S. Sim, S. Lee, and H. Han, 2019: A novel framework of detecting convective initiation combining automated sampling, machine learning, and repeated model tuning from geostationary satellite data. *Remote Sens.*, **11**, 1454, https://doi.org/10.3390/rs11121454.

Héas, P., E. Mémin, N. Papadakis, and A. Szantai, 2007: Layered estimation of atmospheric mesoscale dynamics from satellite imagery. *IEEE Trans. Geosci. Remote Sens.*, **45**, 4087–4104, https://doi.org/10.1109/TGRS.2007.906156.

Heim, N., and J. Avery, 2019: Adaptive anomaly detection in chaotic time series with a spatially aware echo state network. https://arxiv.org/abs/1909.01709.

Hsu, W., and A. Murphy, 1986: The attributes diagram: A geometrical framework for assessing the quality of probability forecasts. *Int. J. Forecasting*, **2**, 285–293, https://doi.org/10.1016/0169-2070(86)90048-8.

Kumler-Bonfanti, C., J. Stewart, D. Hall, and M. Govett, 2020: Tropical and extratropical cyclone detection using deep learning. *J. Appl. Meteor. Climatol.*, **59**, 1971–1985, https://doi.org/10.1175/JAMC-D-20-0117.1.

Lagerquist, R., 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, D. Gagne, and T. Smith, 2020: Deep learning on three-dimensional multiscale data for next-hour tornado prediction. *Mon. Wea. Rev.*, **148**, 2837–2861, https://doi.org/10.1175/MWR-D-19-0372.1.

——, D. Turner, I. Ebert-Uphoff, J. Stewart, and V. Hagerty, 2021: Using deep learning to emulate and accelerate a radiative-transfer model. *J. Atmos. Oceanic Technol.*, **38**, 1673–1696, https://doi.org/10.1175/JTECH-D-21-0007.1.

Lee, S., H. Han, J. Im, E. Jang, and M. Lee, 2017: Detection of deterministic and probabilistic convection initiation using Himawari-8 Advanced Himawari Imager data. *Atmos. Meas. Tech.*, **10**, 1859–1874, https://doi.org/10.5194/amt-10-1859-2017.

Lee, Y., C. Kummerow, and I. Ebert-Uphoff, 2021: Applying machine learning methods to detect convection using using *Geostationary Operational Environmental Satellite-16 GOES-16* advanced baseline imager ABI data. *Atmos. Meas. Tech.*, **14**, 2699–2716, https://doi.org/10.5194/amt-14-2699-2021.

Lin, S., M. Ke, and C. Lo, 2017: Evolution of landslide hotspots in Taiwan. *Landslides*, **14**, 1491–1501, https://doi.org/10.1007/s10346-017-0816-9.

Liu, Y., Q. Ren, J. Geng, M. Ding, and J. Li, 2018: Efficient patch-wise semantic segmentation for large-scale remote sensing images. *Sensors*, **18**, 3232, https://doi.org/10.3390/s18103232.

Loken, E., A. Clark, M. Xue, and F. Kong, 2019: Spread and skill in mixed- and single-physics convection-allowing ensembles. *Wea. Forecasting*, **34**, 305–330, https://doi.org/10.1175/WAF-D-18-0078.1.

McGovern, A., R. Lagerquist, D. Gagne, G. Jergensen, K. Elmore, C. Homeyer, and T. Smith, 2019: Making the black box more transparent: Understanding the physical implications of machine learning. *Bull. Amer. Meteor. Soc.*, **100**, 2175–2199, https://doi.org/10.1175/BAMS-D-18-0195.1.

Mecikalski, J., and K. Bedka, 2006: Forecasting convective initiation by monitoring the evolution of moving cumulus in daytime GOES imagery. *Mon. Wea. Rev.*, **134**, 49–78, https://doi.org/10.1175/MWR3062.1.

——, ——, S. Paech, and L. Litten, 2008: A statistical evaluation of GOES cloud-top properties for nowcasting convective initiation. *Mon. Wea. Rev.*, **136**, 4899–4914, https://doi.org/10.1175/2008MWR2352.1.

——, J. Williams, C. Jewett, D. Ahijevych, A. LeRoy, and J. Walker, 2015: Probabilistic 0–1-h convective initiation nowcasts that combine geostationary satellite observations and numerical weather prediction model data. *J. Appl. Meteor. Climatol.*, **54**, 1039–1059, https://doi.org/10.1175/JAMC-D-14-0129.1.

Mittermaier, M., 2021: A ''meta'' analysis of the fractions skill score: The limiting case and implications for aggregation. *Mon. Wea. Rev.*, **149**, 3491–3504, https://doi.org/10.1175/MWR-D-18-0106.1.

——, N. Roberts, and S. Thompson, 2013: A long-term assessment of precipitation forecast skill using the fractions skill score. *Meteor. Appl.*, **20**, 176–186, https://doi.org/10.1002/met.296.

Molina, M., D. Gagne, and A. Prein, 2021: A benchmark to test generalization capabilities of deep learning methods to classify severe convective storms in a changing climate. *Earth Space Sci.*, **8**, e2020EA001490, https://doi.org/10.1029/2020EA001490.

Mueller, C., and J. Wilson, 1989: Evaluation of the TDWR aviation nowcasting experiment. *Conf. on Radar Meteorology*, Tallahassee, FL, Amer. Meteor. Soc., 224–227.

——, ——, and N. Crook, 1993: The utility of sounding and mesonet data to nowcast thunderstorm initiation. *Wea. Forecasting*, **8**, 132–146, https://doi.org/10.1175/1520-0434(1993)008<0132:TUOSAM>2.0.CO;2.

——, T. Saxen, R. Roberts, J. Wilson, T. Betancourt, S. Dettling, N. Oien, and J. Yee, 2003: NCAR auto-nowcast system. *Wea. Forecasting*, **18**, 545–561, https://doi.org/10.1175/1520-0434(2003)018<0545:NAS>2.0.CO;2.

Qian, X., and H. Wang, 2021: Evaluation of different storm parameters as the proxies for gridded total lightning flash rates: A convection-allowing model study. *Atmosphere*, **12**, 95, https://doi.org/10.3390/atmos12010095.

Roberts, N., and H. Lean, 2008: Scale-selective verification of rainfall accumulations from high-resolution forecasts of convective events. *Mon. Wea. Rev.*, **136**, 78–97, https://doi.org/10.1175/2007MWR2123.1.

Roberts, R., and S. Rutledge, 2003: Nowcasting storm initiation and growth using *GOES-8* and WSR-88D data. *Wea. Forecasting*, **18**, 562–584, https://doi.org/10.1175/1520-0434(2003)018<0562:NSIAGU>2.0.CO;2.

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

Ronneberger, O., P. Fischer, and T. Brox, 2015: U-net: Convolutional networks for biomedical image segmentation. *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, Munich, Germany, Technical University of Munich, https://doi.org/10.1007/978-3-319-24574-4_28.

Sadeghi, M., P. Nguyen, K. Hsu, and S. Sorooshian, 2020: Improving near real-time precipitation estimation using a U-net convolutional neural network and geographical information. *Environ. Modell. Software*, **134**, 104856, https://doi.org/10.1016/j.envsoft.2020.104856.

Sha, Y., D. Gagne, G. West, and R. Stull, 2020a: Deep-learning-based gridded downscaling of surface meteorological variables in complex terrain. Part I: Daily maximum and minimum 2-m temperature. *J. Appl. Meteor. Climatol.*, **59**, 2057–2073, https://doi.org/10.1175/JAMC-D-20-0057.1.

——, ——, ——, and ——, 2020b: Deep-learning-based gridded downscaling of surface meteorological variables in complex terrain. Part II: Daily precipitation. *J. Appl. Meteor. Climatol.*, **59**, 2075–2092, https://doi.org/10.1175/JAMC-D-20-0058.1.

Shi, X., Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, 2015: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Adv. Neural Info. Proc. Syst.*, NIPS, 8 pp., https://papers.nips.cc/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf.

Sieglaff, J., L. Cronce, W. Feltz, K. Bedka, M. Pavolonis, and A. Heidinger, 2011: Nowcasting convective storm initiation using satellite-based box-averaged cloud-top cooling and cloud-type trends. *J. Appl. Meteor. Climatol.*, **50**, 110–126, https://doi.org/10.1175/2010JAMC2496.1.

Sobash, R., J. Kain, D. Bright, A. Dean, M. Coniglio, and S. Weiss, 2011: Probabilistic forecast guidance for severe thunderstorms based on the identification of extreme phenomena in convection-allowing model forecasts. *Wea. Forecasting*, **26**, 714–728, https://doi.org/10.1175/WAF-D-10-05046.1.

Sønderby, C., and Coauthors, 2020: MetNet: A neural weather model for precipitation forecasting. https://arxiv.org/abs/2003.12140.

Starzec, M., C. Hometer, and G. Mullendore, 2017: Storm Labeling in Three Dimensions (SL3D): A volumetric radar echo and dual-polarization updraft classification algorithm. *Mon. Wea. Rev.*, **145**, 1127–1145, https://doi.org/10.1175/MWR-D-16-0089.1.

Stengel, K., A. Glaws, D. Hettinger, and R. King, 2020: Adversarial super-resolution of climatological wind and solar data. *Proc. Natl. Acad. Sci. USA*, **117**, 16 805–16 815, https://doi.org/10.1073/pnas.1918964117.

Sun, W., J. Zhang, J. Meng, and Y. Liu, 2019: Sea surface temperature characteristics and trends in China offshore seas from 1982 to 2017. *J. Coast. Res.*, **90**, 27–34, https://doi.org/10.2112/SI90-004.1.

Walker, J., W. MacKenzie, J. Mecikalski, and C. Jewett, 2012: An enhanced geostationary satellite-based convective initiation algorithm for 0–2-h nowcasting with object tracking. *J. Appl. Meteor. Climatol.*, **51**, 1931–1949, https://doi.org/10.1175/JAMC-D-11-0246.1.

Weusthoff, T., F. Ament, M. Arpagaus, and M. Rotach, 2010: Assessing the benefits of convection-permitting models by neighborhood verification: Examples from MAPD-PHASE. *Mon. Wea. Rev.*, **138**, 3418–3433, https://doi.org/10.1175/2010MWR3380.1.

Wilson, J., and C. Mueller, 1993: Nowcasts of thunderstorm initiation and evolution. *Wea. Forecasting*, **8**, 113–131, https://doi.org/10.1175/1520-0434(1993)008<0113:NOTIAE>2.0.CO;2.

Zhou, Z., M. Siddiquee, N. Tajbakhsh, and J. Liang, 2019: Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Trans. Med. Imaging*, **39**, 1856–1867, https://doi.org/10.1109/TMI.2019.2959609.