Predicting Tropical Cyclone Formation with Deep Learning

Quan Nguyen[1] and Chanh Kieu[1,*]

[1]*Department of Earth and Atmospheric Sciences, Indiana University, Bloomington, IN 47405*

Corresponding author: *Chanh Kieu, Department of Earth and Atmospheric Sciences. Indiana University, Bloomington, IN 47405. Email: ckieu@indiana.edu.

ABSTRACT: Exploring new techniques to improve the prediction of tropical cyclone (TC) formation is essential for operational practice. Using convolutional neural networks, this study shows that deep learning can provide a promising capability for predicting TC formation from a given set of large-scale environments at certain forecast lead times. Specifically, two common deep-learning architectures including the residual net (ResNet) and UNet are used to examine TC formation in the Pacific Ocean. With a set of large-scale environments extracted from the NCEP/NCAR reanalysis during 2008-2021 as input and the TC labels obtained from the best track data, we show that both ResNet and UNet reach their maximum forecast skill at the 12-18 hour forecast lead time. Moreover, both architectures perform best when using a large domain covering most of the Pacific Ocean for input data, as compared to a smaller subdomain in the western Pacific. Given its ability to provide additional information about TC formation location, UNet performs generally worse than ResNet across the accuracy metrics. The deep learning approach in this study presents an alternative way to predict TC formation beyond the traditional vortex-tracking methods in the current numerical weather prediction.

SIGNIFICANCE STATEMENT: This study presents a new approach for predicting tropical cyclone (TC) formation based on deep learning (DL). Using two common DL architectures in visualization research and a set of large-scale environments in the Pacific Ocean extracted from the reanalysis data, we show that DL has an optimal capability of predicting TC formation at the 12-18 hour lead time. Examining the DL performance for different domain sizes shows that the use of a large domain size for input data can help capture some far-field information needed for predicting TCG. The DL approach in this study demonstrates an alternative way to predict or detect TC formation beyond the traditional vortex-tracking methods used in the current numerical weather prediction.

1. Introduction

The life cycle of a tropical cyclone (TC) is typically divided into six stages including genesis, tropical disturbance, tropical depression, tropical storm, hurricane, and dissipation. Among these six, the genesis stage (typically 2-5 days) during which a weak atmospheric disturbance grows into a mesoscale tropical depression with a close isobar and the maximum surface wind $> 17ms^{-1}$ is perhaps the most difficult to forecast because of its unorganized structure and ill-defined TC characteristics (Karyampudi and Pierce 2002; Houze 1982; Kieu and Zhang 2009; Hennon et al. 2011, 2013; Vu et al. 2021; Tien et al. 2020). For this genesis period, synergetic interactions among various dynamical and thermodynamic processes at different scales may eventually result in the generation of a self-sustained, warm-core vortex before subsequent intensification can proceed. This early TC formation process is intricate and highly nonlinear that no single mechanism could operate in all ocean basins, rendering tropical cyclogenesis (TCG) forecast challenging in practice. To date, this multi-faceted nature of TCG is the main obstacle that prevents one from understanding and predicting TCG in real time.

Recent advancements in machine learning (ML) have sparked more interest in using ML for meteorological problems. Broadly speaking, ML is a technique that allows one to find patterns (features) and make predictions without knowing all details of physical and/or dynamical principles underlying the data (Murphy 2012; Hastie et al. 2009). By training an ML model over a large number of data, specific features corresponding to a given set of classifiers/labels can be detected with different accuracy and interpretability, depending on which supervised or unsupervised methods are used. With recent advances in hardware architecture and algorithms, various ML models have been developed and optimized to efficiently process large datasets. This rapid development of ML techniques opens up many potential applications of ML to a wide range of research and practical problems as discussed in, e.g., Murphy (2012); Hastie et al. (2009); Fenner (2019).

While ML techniques have been increasingly applied to different areas in atmospheric science, the applications of ML specifically to TC research are relatively new and preliminary. Most of the recent studies on the use of ML techniques for TC research focused on the analyses of satellite images to improve the track and intensity forecasts of an existing TC or classify TC evolution based on different pre-existing cloud patterns. For example, using the observations of surface precipitation rate, the total water content, and the tropopause temperature from the TRMM satellite

products, Su et al. (2020) compared the performance of several different ML schemes such as logistic regression, random forecast, and decision tree. Their results showed that these variables are approximately correlated with the subsequent 24-hour TC intensity change, and thus can be used as predictors for TC intensity forecast. Likewise, Miller et al. (2017) used deep learning with GOES IR satellite datasets to train a convolutional neural network, which can search for cloud patterns and categorize TC intensity based on different cloud shapes of tropical disturbances. This line of ML approach has been further advanced to help improve TC forecasts by integrating the tracking information and/or other reanalysis data, with some promising performance for TC nowcasting and forecasts (Gao et al. 2018; Kim et al. 2019; Giffard-Roisin et al. 2020).

Along with TC classification and track/intensity forecast, a recent study by Zhang et al. (2019) proposed an approach that employs a set of TCG predictors to train several different ML classifiers. Their experiments with a range of ML classifiers showed that the Adaboost approach appears to be the most effective in capturing TC formation from mesoscale convective systems (MCS), as compared to the traditional approach based on the genesis potential index (GPI). The better performance of Adaboost is seen in all basins and forecast lead times from 6 to 48 hrs, suggesting the potential applicability of boosting iterative ensemble training in capturing TCG associated with some pre-existing MCSs. Another approach of ML for TCG prediction is to use satellite images of precursor clouds (often recorded as Invests in operational forecasts) and classify which ones will develop into a TC at a later time (Zhang et al. 2015; Park et al. 2016; Matsuoka et al. 2018; Kim et al. 2019). In this approach, TC precursor signals, which are often manifested in terms of cloud or radiance, must be given in advance such that the analyses centered on these cloud clusters can be carried out. Using different classifiers such as decision trees, random forest, or support vector machine approaches, these TC images can be then classified into developing or non-developing systems. In all of these above studies, it is essential to obtain and train an ML algorithm on a set of images with some existing TC-related cloud signals.

While the ML classification approach could be customized for predicting TCG as mentioned above, predicting TCG based on scalar predictors such as the area-averaged 850 hPa vorticity, low-level humidity, wind shear, or potential genesis index is generally insufficient. Various observational and climatological studies on TCG showed that the area-averaged favorable conditions for TCG do not guarantee that a TC would form (McBride and Zehr 1981; Gray 1998; DeMaria et al. 2001;

Emanuel and Nolan 2004a; Camargo et al. 2014; Peng et al. 2012; Halperin et al. 2013; Tang et al. 2020). In fact, there are many different pathways for TCG that area-averaged predictors cannot fully capture. For example, TCG predictors would not allow for taking into account environmental asymmetries or other local signals that can help spin up TC circulations in different basins. In this regard, the better performance of ML classifiers relative to the traditional genesis index benchmark forecast as presented in, e.g., Zhang et al. (2019) may not be very useful for examining different TCG mechanisms or large-scale environmental asymmetries.

Given that ML classifiers based on spatially-averaged TCG predictors do not directly take into account the spatial distribution of the environment where TCs form, how to employ ML methods to study different TCG pathways in real atmospheric conditions when *there exist no clear or pre-existing TC signals in advance* is still a challenging question. In this study, we present an ML framework for TCG prediction, based on the convolutional neural network (CNN) method for gridded meteorological datasets. Our main objective here is to explore how CNNs can take into account not only different environmental factors relevant to TCG but also the spatial distributions of these factors at different forecast lead times via convolution. By further examining different domain sizes of input data, we can also quantify how remote and local environments influence TCG prediction and its accuracy beyond the traditional classification approaches. We wish to emphasize that our focus in this study is on predicting the early TC formation stage before any TC signal appears. As such, traditional classifications or common vortex tracking methods that directly detect a TC vortex from gridded dataset cannot be directly applied during the TCG period as discussed in, e.g., Tien et al. (2020).

The rest of this work is organized as follows. In the next section, details of our CNN algorithms and feature selection processes are presented. An approach to monitor and evaluate the performance of CNN for TCG prediction will also be discussed. Section 3 presents the detailed results for two CNN methods examined in this study, and Section 4 provides sensitivity analyses for our methods. A summary and concluding remarks are then given in the final section.

2. Methodology

a. Deep learning approach

Among many different methods for image processing, deep learning (DL) has become increasingly popular due to its ability to search for possible signals of any feature from a large input dataset. A major building block of CNN-based deep learning is convolutional layers, which act as a filter to inputs with different activation functions. A sequence of the application of CNN kernels (or filters) in deep learning results in the so-called feature maps that can capture the shape, strength, and possibly the location of key features from input images. Note that any detected feature is highly tailored to the input labels (targets) that one feeds to the algorithm for supervised learning. As such, proper labeling is required so that supervised deep learning can be effective for feature extraction tasks. Because of this capability, DL has a wide range of applications in image recognition, classifications, object segmentation, or face recognition. With the goal of searching for environmental features that are favorable for TCG within a given domain, CNN-based deep learning techniques are thus naturally suitable for the TCG problem.

In applying CNN to predicting TCG in operational practice, a challenging issue is that there is no apparent signal of a TC vortex within the domain at a given forecast time. Recall that the key advantage of CNN is to detect a labeled feature in input data by optimizing a set of kernel weights. With a well-designed architecture of convolution layers [1], one can extract a feature anywhere within the domain (often known as translation equivariance, Goodfellow et al. (2017)). This exact advantage of CNN, however, also makes it hard to apply directly to the TCG prediction problem, as we have to predict in advance 1) whether a TC will develop before it even exists, and 2) where the TC will form inside the image at some given forecast lead time. Until a tropical disturbance (also known as Invest in the operation) is identified, no information on TC location or strength is reported. Without a clear signal of TCs from input data, the application of CNN to TCG prediction is therefore subtle in practice because it now requires a different approach and interpretation beyond the traditional classification problems.

Given such unique characteristics of TCG prediction, we will approach this problem by first hypothesizing in this study that the necessary ingredients for TCG can be detected from the

[1] A good ML model is a subjective concept that depends on each application. In the traditional sense of machine learning, a good model for classification should have an accuracy above 80%, using a test and/or validation dataset.

ambient environment by DL convolution at some given forecast lead times. This hypothesis is supported by previous modeling and observational studies on TCG, which suggested several key environmental conditions for TCG such as warm sea surface temperature, low vertical shear, moist lower troposphere (see, e.g., Gray 1998; Emanuel and Nolan 2004b; Kieu and Zhang 2008; Nolan et al. 2007; Camargo et al. 2014; Tang et al. 2020; Kieu et al. 2023). By training a DL model on a set of input data and its corresponding TCG labels at different forecast lead times, it is expected that CNN can capture hidden environments needed for TCG and allow for skillful TCG prediction. We note again that convolution is essentially an operator that acts as a spatial filter of all irrelevant environmental features within the input domain. Although we do not know exactly what features will be retained for TCG prediction, convolution kernels naturally take into account the spatial distribution of the ambient environment that classification models based on predictors could not.

With this hypothesis, we consider next a set of meteorological variables critical for TCG as different channels of an input image and examine how these multi-channel images can capture TCG at different forecast lead times. In this study, two popular DL architectures will be examined. The first is an algorithm known as residual neural network (ResNet), which was proposed by He et al. (2015) to help address the vanishing gradient issue with deep neural network models. Specifically, a skip connection between two consecutive convolution blocks was introduced to alleviate the problem of vanishing gradient. These skip connections form a highway to allow gradient information to flow from the output layer to the very first layer without losing information about the gradient function, thus enabling deeper neural network training with higher accuracy (He et al. 2015).

Among many different variants of ResNet, we found that the 18-layer ResNet (see Fig. 1 achieves the best results for our dataset. In this design, each convolution block will progressively reduce the spatial dimensions while increasing the depth of output feature maps. This configuration allows the deep neural network to have a larger receptive field in later blocks and consequently more meaningful feature maps in later stages. In addition, skip connections are introduced after every two convolution layers to facilitate the highway for the gradient to flow to the very first layer, effectively reducing the gradient-vanishing problem.

Because the predictions made by ResNet are limited only to whether or not a TC would form at a certain lead time without any information about where the location of TC formation would be, a

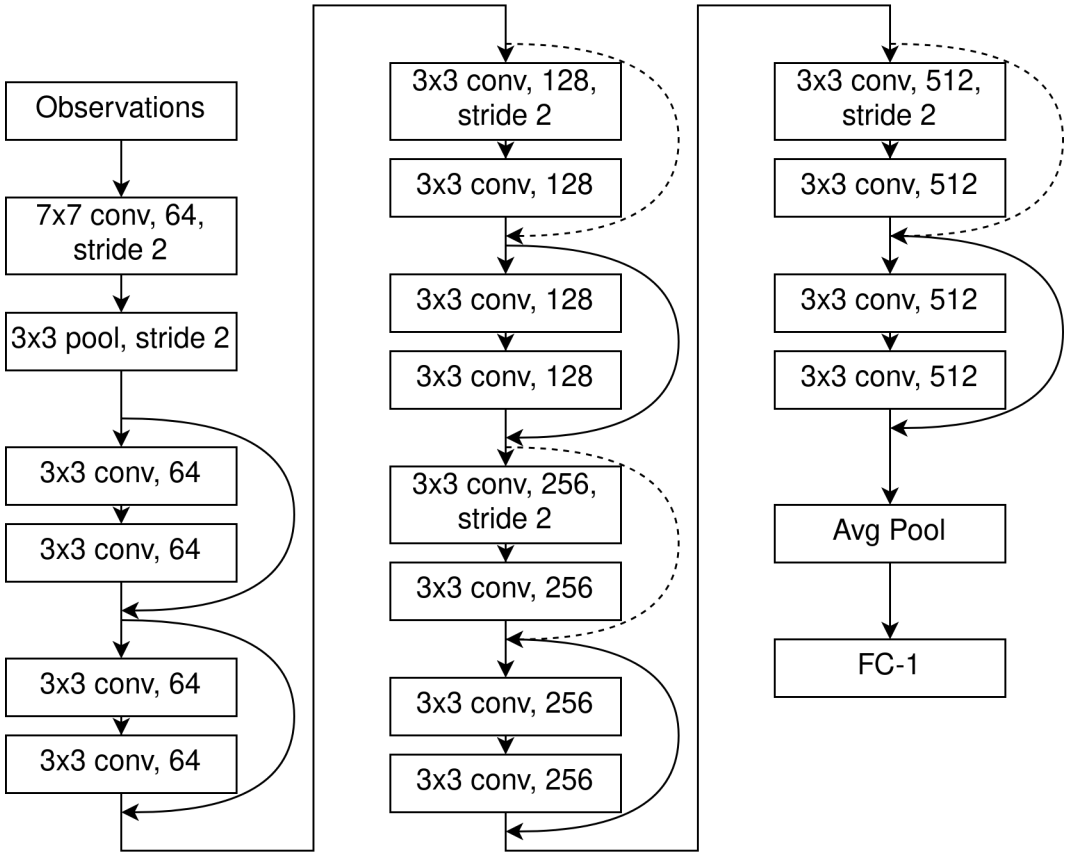

FIG. 1. The architecture of the ResNet-18 model that is designed for predicting TC formation in this study, whose input (i.e., the "Observations" block) may include gridded climate data, numerical model forecast output, or satellite imagery. Note that the curved arrows denote the skipped step in our ResNet design, and the last block (fully-connected, or FC-1) is the yes/no forecast of a TCG event, the dashed curved arrows denote the skipped step with 1x1 convolution layers to match the spatial dimensions of the next convolution block.

second DL architecture, known as UNet model (Ronneberger et al. 2015), is used to provide further the probability distribution of TCG at every point in the domain. UNet was originally designed for biomedical image processing, in which the model has to learn to recognize which pixels belong to a cell. A typical UNet architecture consists of an encoder and a decoder branch as shown in Fig. 2. The encoder branch progressively compresses and transforms original images into compact vector representations, while the decoder decodes and transforms the compact information into useful predictions.

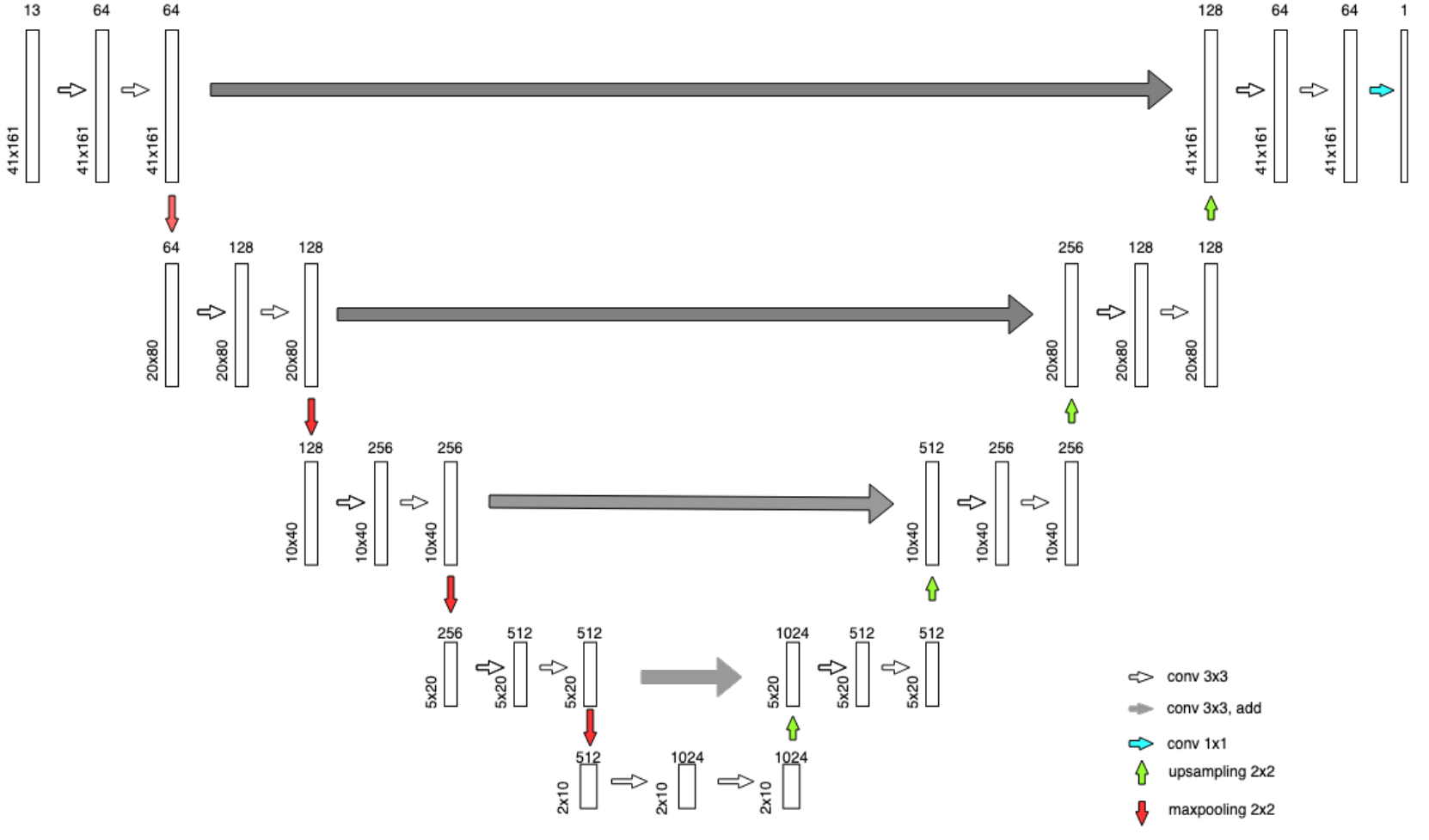

FIG. 2. Similar to Figure 1 but for the UNet architecture, with an output given as a probability distribution map instead of yes/no prediction. The red/green arrows denote the maxpooling and upsampling steps, while the white/gray arrows denote the convolutional steps.

For our TCG application, both the encoder and decoder branches of the UNet model consist of 5 convolution blocks. In the encoder branch, each block has two 3x3 convolution layers followed by a 2x2 max-pooling layer to reduce the spatial dimensions of feature maps by half, which will then be fed to the next block. The final convolution block in the encoder has one 3x3 convolution layer to produce a compact tensor of shape 2x10x1024. Similar to ResNet, the higher layers in the encoder branch have larger receptive fields, thus capable of encoding large-scale environmental conditions. The output of the encoder branch is then fed to the decoder branch. Note that each block in the decoder branch has two 3x3 convolution layers followed by a 2x2 upsampling layer to gradually increase the spatial dimensions to match the final target density probability map. In addition to receiving input from the previous block, each block also receives additional input from the corresponding convolution block in the encoder block represented as gray arrows in Fig. 2. This additional input provides fine-scale information for the decoder, while acting as a shortcut for gradient flows and preventing the gradient vanishing problem. Therefore, our UNet architecture facilitates information flow from both local and large-scale environmental factors to produce predictions for each grid point in the final density map.

The choice of the loss function and optimizer is also important to the performance of deep learning models, especially when processing a large amount of data during the training process. For ResNet, we use the sigmoid focal loss (Lin et al. 2017), which is known to enable deep models to learn effectively in an imbalanced dataset context. For the UNet model, we use a common loss function for the image segmentation problem known as dice loss (Eq. (1)).

$$\text{Dice Loss} = 1 - \frac{\sum_i^N p_i g_i}{\sum_i^N (p_i + g_i)}, \tag{1}$$

where p_i and g_i are the predicted probability and the true probability, respectively. For both models, an adaptive gradient descent algorithm (Kingma and Ba 2014) is used to train the models.

b. Data

To train our DL models, the NCEP final analysis (FNL) dataset at a horizontal resolution of 0.5 degrees during 2008-2021 was used. Our area of focus in this study is the North Pacific Ocean during the main TC season from May to November, as this is the most active ocean for TC activities.

While this NEP/FNL data is global, we examine in this study only two data domains. The first data domain is from [5°N-35°N] × [100°E-100°W] that covers most of the North Pacific tropical region. The second smaller domain ([5°N-20° N] × [100°E-140°E]) covers a sub-area within the northwestern Pacific basin. These two different data domain sizes are needed so we can evaluate how the different data domain sizes could change the performance of our DL models for TCG prediction.

For both domains, the same 13 meteorological variables most relevant to TCG processes were extracted from the FNL data and then treated as different channels of input data for our DL models (see Table 1 for the list of these variables). While these variables were chosen based on their potential impacts on TCG as shown in the previous studies (see, e.g., Hill and Lackmann 2011; Nolan et al. 2007; Ferrara et al. 2017; Camargo et al. 2014; Kieu and Zhang 2018; Vu et al. 2021), how effective they are within the DL framework or their relative importance in detecting TCG in the Pacific Ocean at different forecast lead times is not fully understood. Note that one can in principle include any other variables such as latent heating, convective precipitation, cloud types, or total water content to improve the performance of DL models. However, our main goal in this study is to present an efficient DL model that can be easily used with the current global GFS input data or climate projection output such that the model is as broad and general for different input data types as possible. Thus, we limit our input channels to the 13 variables listed above to speed up our training, with an underlying assumption that other relevant variables are cross-dependent and will be captured via convolution neural networks.

Among those 13 variables, we note that absolute vorticity is a diagnostic variable derived from horizontal winds, and so it should be inherently accounted for by the wind information during the training process. Due to its important role in the TCG process, the direct inclusion of this variable could however help improve the performance of our DL models as compared to a simple use of horizontal winds only (a process known as feature engineering in ML). Unlike the traditional vortex tracking algorithm that detects potential TCG locations by searching for a local high vorticity center, DL models process the global distribution of vorticity to identify TCG locations. As such, it avoids the issues of irrelevant local centers that traditional vortex tracking algorithms often encounter.

To create TCG labels, the International Best Track Archive for Climate Stewardship (IBTrACS) (Knapp et al. 2010) was used to label all TCG events and locations. In this work, a TCG event is

Variable	Pressure Levels
Absolute Vorticity	900mb, 700mb
Relative Humidity	750mb
Temperature	900mb, 500mb
Geopotential Height	500mb
Vertical Wind	500mb
U-wind	800mb, 200mb
V-wind	800mb, 200mb
CAPE surface	-
Surface Temperature	-

TABLE 1. Variables extracted from the NCEP/FNL data that are fed into deep learning models in this study.

defined as the very first time a storm was recorded in the best track data. With this definition, we can scan through all TC track records and take the first recorded location of each storm in each domain to create a target output for a TCG event. In addition, all the dates and times for which several TCs co-existed in the IBTrACS were filtered out to avoid miss labeling the pre-existing TCs as a TCG event, using the procedure described in Nguyen (2023). Finally, all relevant information related to a TCG event including its longitudes, latitudes, date, and time was stored in a csv database to facilitate our data sharing and input to the DL interface. This pre-process workflow is provided in our open-source version control Github listed in the Acknowledgement section.

With these pre-processed input datasets, we followed the standard protocol in DL models and split the data into 3 different subsets including training, validation, and testing. Specifically, the data from 2008-2014 was used for training, while data from 2015-2017 was reserved for validation. The remaining data were then used for testing. Note that this TCG dataset is highly unbalanced in the sense that most of the input data ($> 80\%$) contain no TCG events. This is a challenging issue for designing a DL algorithm for TCG prediction. Our approach to this unbalanced data problem is to generate a subset of input data with augmentation such that the number of TCG events (positive labels) is about a quarter of the total input data during the training. By maintaining a 1:4 ratio for the TCG dataset and repeating the training process for different sampling, we can evaluate the robustness of our DL model. Figure 3 summarizes the overall pipeline architecture of our DL models and the corresponding data flow. For this workflow, we normalize and standardize the datasets at each level to help the learning process be more efficient because the input variables have different ranges and units at different pressure levels. Due to limited data on TCG events, the

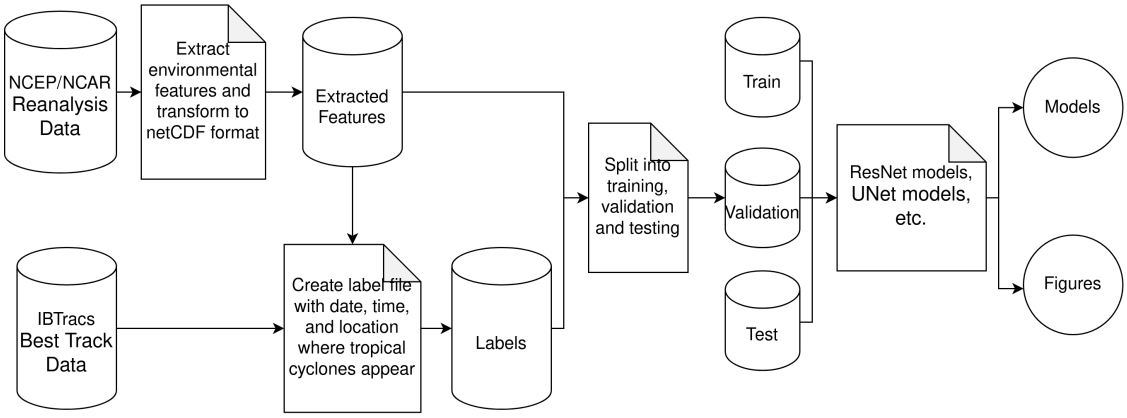

FIG. 3. A complete design of the deep learning framework for TCG prediction used in this study, which shows the workflow starting from meteorological data inputs to the final output.

common early-stopping strategy was also used to prevent the model from overfitting the training data.

It is worth mentioning here that the use of the NCEP/FNL data would not prevent our models from being applied to other datasets. This is because ML generally learns key features from any input data, so long as the data contains the features matching with assigned labels. Learning from the NCEP/FNL reanalysis dataset can be therefore treated as preliminary learning, from which one obtains some preliminary information about the key environmental features for TCG. Our ML models can be then improved further by adding more data from other global or climate models later on, which refine the DL models for different applications such as climate projection or real-time forecast. This process, often known as transfer learning in ML applications, can help save the training process of future ML models, which may take a very long time to train on large datasets. Since the NCEP/FNL data reflects a good degree of large-scale observation, training ML models on this dataset will help short-cut future training with different datasets in case one can re-use our model weights obtained from the NCEP/FNL data. From this perspective, training ML models with NCEP/FNL data is a necessary step rather than a limitation of our models, which we will discuss in more detail in the Result section.

c. Integrated Gradients

As expected from any DL development, it is important to understand what CNNs learn from input data and how they apply the knowledge for prediction instead of running a DL model as a black box. There are several techniques for this purpose based on, e.g., intermediate output visualization, heatmap, or filter visualization. In this study, we follow an approach that is based on integrated gradient (IG, see Sundararajan et al. (2017)) to gain some insights into the performance of our DL models. Recall that ResNet produces yes/no predictions based on features in the input without letting us know where it obtains its information for prediction. Using the IG analyses, it is possible to understand further how a DL model makes use of input data for its decision.

Specifically in this study, we use the IG expression defined for a function $f(x)$ as follows

$$\text{Integrated Gradient}(x) = (x - x') \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x} d\alpha \tag{2}$$

where $f(.)$ is the ResNet or UNet model, x is the input we want to diagnose, and x' is a reference input such that $f(x') = 0$. For our analyses, the reference input x' is chosen to be all 0, and the implementation of IG is based on the Tensorflow API (Abadi et al. 2015) [2]. With the above IG, we can then produce spatial maps that show what regions of an image are used by a DL model to produce a forecast.

d. Validation Metrics

For categorical forecasts like TCG prediction, there are a number of different metrics to evaluate the performance of DL models. In this study, we use three key metrics including Recall, Prediction, and F1 score derived from the confusion matrix to evaluate our DL models. This confusion matrix (also known as a categorical or contingency table in the traditional weather forecast) displays the number of correct predictions, hit rejections, false alarms, and misses in categorical forecasts. Physically, Recall shows how well an ML algorithm can detect positive cases, which is given by

$$\text{Recall (R)} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3}$$

[2]In Tensorflow, the gradient of an output of a model with respect to the input can be easily calculated using "tf.GradientTape".

A higher Recall would correspond to a more correct prediction of TCG events as compared to the number of missed events, (often referred to as the probability of detection (POD) in the categorical weather verification). Precision, on the other hand, represents how accurate the positive predictions of the algorithm are and is defined as follows:

$$\text{Precision (P)} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4}$$

As shown from the above definition, Precision is essentially a complement of the false alarm rate (FAR) in the sense that P = 1 - FAR, which is more commonly known in the categorical weather verification as a success ratio. Generally, R and P provide different information about the model performance that may however trade-off. To combine these scores into a single effective metric, F1 score is introduced to assess quickly the overall performance of DL models, which is given by

$$\text{F1} = \frac{2RP}{R+P} \tag{5}$$

A perfect ML model will have $R = P = 1$, and so F1 is equal 1. For an actual ML model, R and P will not in general be equal to 1. Practically, a good DL model for TCG forecast should have R and P at least comparable to the POD or the success ratio in the current operational physical-based models (i.e., $P > 0.5$ and $R > 0.5$). These minimum requirements for R and P ensure that the DL model is at least skillful for practical applications. By examining how R and P vary for a range of forecast lead times, model hyperparameters, or input data types, one can evaluate the capability of DL models for TCG prediction and optimize the models relative to physical-based models as expected.

3. Results

a. ResNet performance

Figure 4a shows first the performance of ResNet in predicting TCG for the large domain covering most of the North Pacific Ocean, using all 13 input variables. As seen in Fig. 4a, ResNet is doing reasonably well with $R > 0.9$ for most forecast lead times, indicating that 90% of the predicted TCG events are correctly detected by ResNet. Similar to TCG prediction directly from global numerical

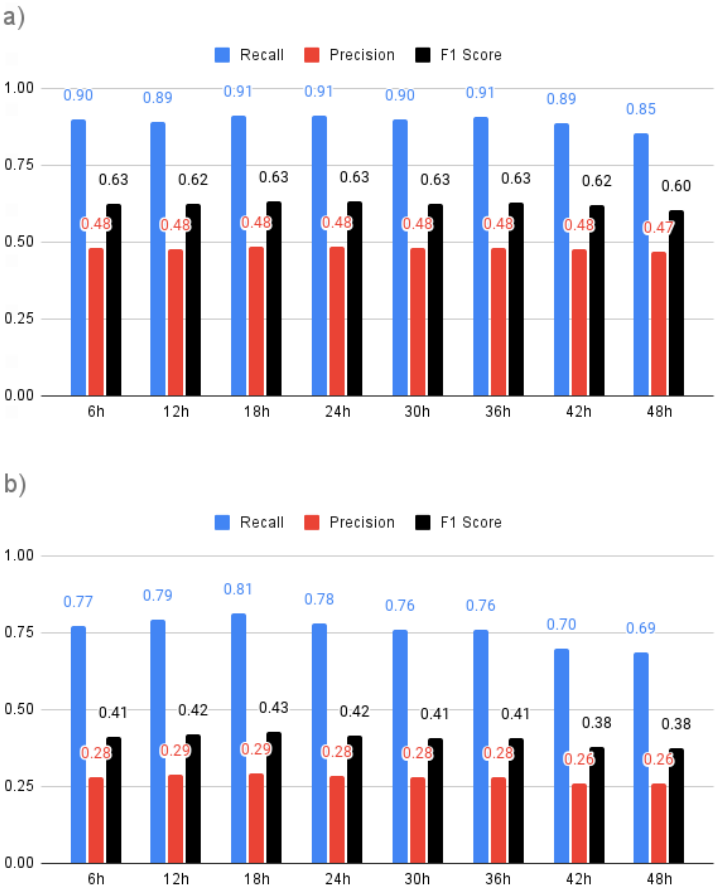

FIG. 4. ResNet's performance including Precision (red), Recall (blue), and F1 score (black) at different forecast lead times for a) a large input domain covering a part of the Northern Pacific Ocean (from $5°N$ to $45°N$ and $100°E$ to $260°E$; and b) a subdomain in the Northwestern Pacific basin.

models, the precision of ResNet is however relatively low ($P \leq 0.5$) at all lead times (i.e., ResNet tends to produce a high false alarm ratio > 0.5). The overall performance of ResNet, which is represented by the F1 score, is optimal at 24-36 hr lead times (≈ 0.63) and gradually decreases as expected for any real-time forecasting systems. That is, a longer forecast lead time would have lower accuracy overall due to the limited predictability of the atmosphere.

At a longer lead time (> 48 hr), we noticed that ResNet starts to behave quite differently, with the loss and validation curves oscillating widely with epochs during the training (not shown). Our attempt to use fewer ResNet layers or input channels could help improve the convergence of the model, which captures a decay of the F1 score with lead time as expected. However, this

performance is no longer comparable as the ResNet input and design have changed, making it hard to compare the results. We speculate that such behavior of ResNet is caused by the vanishing gradient of the model when the TCG signal is not recognizable at a long lead time, but do not have any further evidence to support this speculation. As such, we will limit our analyses of the DL model performance to lead times $\leq$ 48 hours hereinafter.

The fact that ResNet could capture a high recall rate with F1 $>$ 0.63 from 0-48 hrs is noteworthy because it suggests that DL could potentially provide some forecast skill at short lead times, at least for the set of training data used in this study. The implication of this ResNet's performance is non-trivial, because we recall that any prediction from our DL algorithm herein is based purely on a given state without any dynamical or physical principles as in dynamical models. The fact that ResNet could capture such decaying forecast accuracy with forecast lead time suggests that ResNet is able to detect some environmental signals needed for TC development, even without any governing dynamical equations. Of course, the low Precision score also implies that DL tends to have a high false alarm rate due to the generally favorable conditions for TCG most of the time during the main TC season. However, this same issue with a high false-alarm rate is also common among dynamical models, and highlights the key difficulty in predicting TCG that both physical-based and DL-based models currently have to cope with.

While the high recall score from ResNet may appear comparable to the POD score from real-time verification of TCG forecast in the current operational global forecast models (e.g., Henderson and Maloney 2013; Cossuth et al. 2013; Halperin et al. 2013; Li et al. 2016; Yamaguchi and Koide 2017; Halperin et al. 2020), any direct comparison between ResNet and global model forecast should be highly cautioned. This is because the global TCG verifications are inhomogeneous and contain different types of forecast errors. In addition, these global model verifications are generally derived for a range of forecast hours such as 6-120 hrs in Halperin et al. (2013, 2016) instead of each lead time as in our study. Therefore, the ResNet's Recall score and POD from global models are not directly comparable. Despite these differences, that both physical-based and DL models possess similar Recall/POD and high false alarm rate regardless of the ocean basin indicates some inherent limited predictability for the TCG processes, even at a short range lead times.

To further analyze how the performance of ResNet changes with the input data domain size, Fig. 4b shows similar scores using an environment within the subdomain in the Pacific Ocean from

[5-20°N]×[100-140°E]. It is of interest to see from Fig. 4b that using the local environment in this subdomain results in a degraded performance of ResNet in predicting TCG across the metrics and forecast lead times. This degradation of ResNet for the small domain is important from the physical standpoint, because it indicates that local environments inside a smaller domain are insufficient to capture its own TCG. That is, a significant portion of the information required for TCG prediction in one area must be drawn from far-field regions rather than just in the vicinity of a TCG location. This result appears to be consistent with those obtained from previous physical-based modeling studies of TCG, which demonstrated the difficulty in simulating TCG if the model domain is too small (see, e.g., Chen et al. 2012; Goswami and Mohapatra 2014).

Given such sensitivity of ResNet to the input domain size as shown in Fig. 4, it is important to examine why ResNet displays such intriguing performance by using the IG analyses. Specifically, we want to look for where the environmental information used by ResNet to predict TCG comes from and how this information depends on the domain size. For this, the IG analyses given by Eq. (2) for several different true positive examples (i.e., ResNet predicts a "Yes" TCG event, and observation also recorded a TCG event) are shown in Fig. 5, using the large domain input. While ResNet's prediction is correct in these examples, the information used to predict these TCG events comes actually from different sources, thus exposing an issue with the application and interpretation of ResNet for TCG. Specifically for the case of Typhoon Wukong (2018) (Fig. 5a), the most significant information required for its TCG prediction comes from the two blue boxes near the South China Sea and the China East Sea instead of the Central Pacific where Wukong formation occured. A similar issue also occurs for two other TCG cases of Typhoons Mirinae and Nida (2021) (Fig. 5b) for which the most influential information for predicting these two TCG events comes not only from their local environment (i.e., the shaded areas within the orange boxes), but also from a nearby storm close to the Vietnam coastal region (i.e., the shaded area in the blue box). In this regard, these IG analyses help explain why using a smaller domain size degrades the performance of ResNet, mostly because some hidden remote information from the far field is no longer available for its decision.

While the IG analyses could provide some guidance on where ResNet extracts its information for TCG forecast, we note that IG alone is still insufficient to answer a deeper question of what environmental asymmetries play the key role in TCG prediction. Recall that ResNet consists of multiple

layers of convolution applied to its multi-channel input data during the training process. These convolutional layers are further modified via maxpooling layers at every step, which inherently take into account the impacts of all environmental asymmetries to extract the best TCG-related features. For a typical image classification problem with a well-defined object such as a cat or a dog, one could use standard techniques such as heat map, or gradient visualization to see where features are learned. For our TCG forecast problem in which a TC signal is not even apparent at the time of the forecast, finding exactly what environmental asymmetries and their corresponding location within the input domain or channel is more challenging and beyond what IG could answer. All we could learn from the IG analyses is that the information needed for predicting a TCG event comes from certain places within the domain, but not what environmental features are most decisive. In this regard, the question of how spatial asymmetries in the large-scale environment contribute to TCG still cannot be answered in this study.

Regardless of its disadvantage in quantifying environmental features, IG could still highlight that simply looking at the scalar metrics such as F1 scores or Precision when predicting TCG is inadequate for diagnosing the performance of a DL model. Specifically, the information most useful for a TCG forecast might come from unknown features or locations, even though the forecast is categorically correct. In this regard, IG helps uncover ResNet as well as understand how data is used to make a prediction beyond the black-box perception of DL models. Since ResNet does not generally answer the question of *where a TC would form* within the input domain, we examine next the UNet model that can provide us more TCG information.

b. UNet performance

Unlike the ResNet model that provides only yes/no prediction, UNet can provide additional information about where a TCG event would occur, along with corresponding TCG probability. To gain a general sense of how UNet performs, Fig. 6 shows the overall performance of UNet at different forecast lead times for two domain sizes. Similar to ResNet, one notices immediately that the performance of UNet on the large domain outperforms that of the small domain at all forecast lead times. Specifically, UNet displays a peak forecast skill at 12-28 hr with F1 0.21 for the large input domain, which is almost double the F1 score obtained from the small input domain. Regardless of the domain sizes, the performance of UNet is reduced by almost 50% after 48 hr for

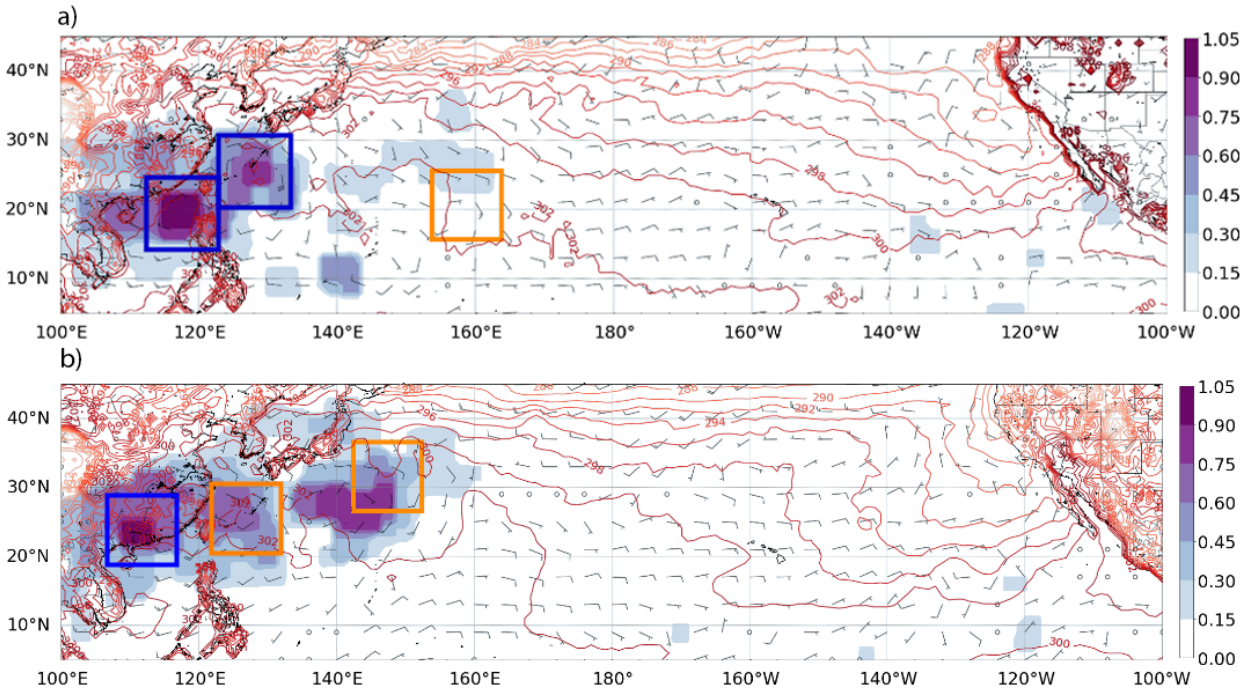

FIG. 5. Horizontal distribution of integrated gradient (shaded) obtained from Resnet's TCG predictions for a) Typhoon Wukong valid at 1800 UTC July 20, 2018 and b) Typhoon Mirinae and Typhon Nida valid at 1200 UTC August 4, 2021. Superimposed are SST (red contours, unit K) and the corresponding wind barfs at 850 hPa. The orange boxes show the observed TCG locations while the blue boxes highlight the remote locations that are decisive to ResNet's TCG prediction.

all metrics, thus confirming the deteriorated forecast skill for longer forecast lead times similar to that observed in dynamical models.

To see how UNet could deliver the prediction of both the probability and the location of TCG, Fig. 7 shows an example of a true positive case for which UNet could correctly predict the expected formation of Typhoon Chanthu (2021), along with the probability distribution of Chanthu's genesis event. One can see that UNet could indeed capture not only the probability of Typhoon Chanthu formation but also the corresponding location of its cyclogenesis event as expected. In this regard, UNet could provide more information for TCG prediction beyond a simple yes or no prediction as for ResNet.

It is of interest to note however that UNet has significantly worse performance than ResNet across metrics for both the large and small domains. While ResNet could reach an F1 score of 0.63 for

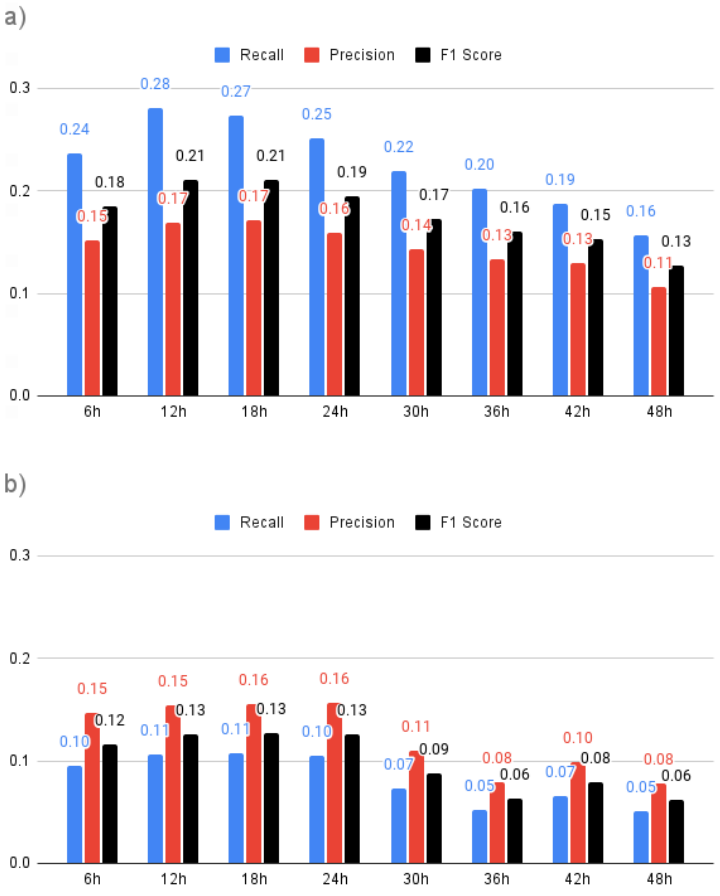

FIG. 6. Similar to Fig. 4 but for the UNet model with two different domain sizes: a) a large domain over the north Pacific Ocean, and b) a small domain within the northwest Pacific basin.

18-36 hr lead times, the maximum F1 score that UNet can achieve is just 0.21 as shown in Fig. 6b. Similarly, F1 is much lower if the small domain is fed to UNet, with the maximum F1 score of only 0.13 during 12-24 hr lead time. Such a much weaker performance of UNet as compared to ResNet is the trade-off that one would have if more information on TCG prediction is extracted from the input data, which is caused by UNet's complicated architecture and outputs.

This trade-off can be best seen in an example of Storm 01E (2018) shown in Fig. 8. For this case, UNet could predict correctly a true positive prediction in terms of yes/no TCG event as expected, yet the location of the 01E's genesis is very different from that of the real TCG event. Apparently, if one simply uses the yes/no categorical validation, the performance of UNet would be perfect. However, if the point-like probability evaluation is applied at each grid point, then UNet fails to

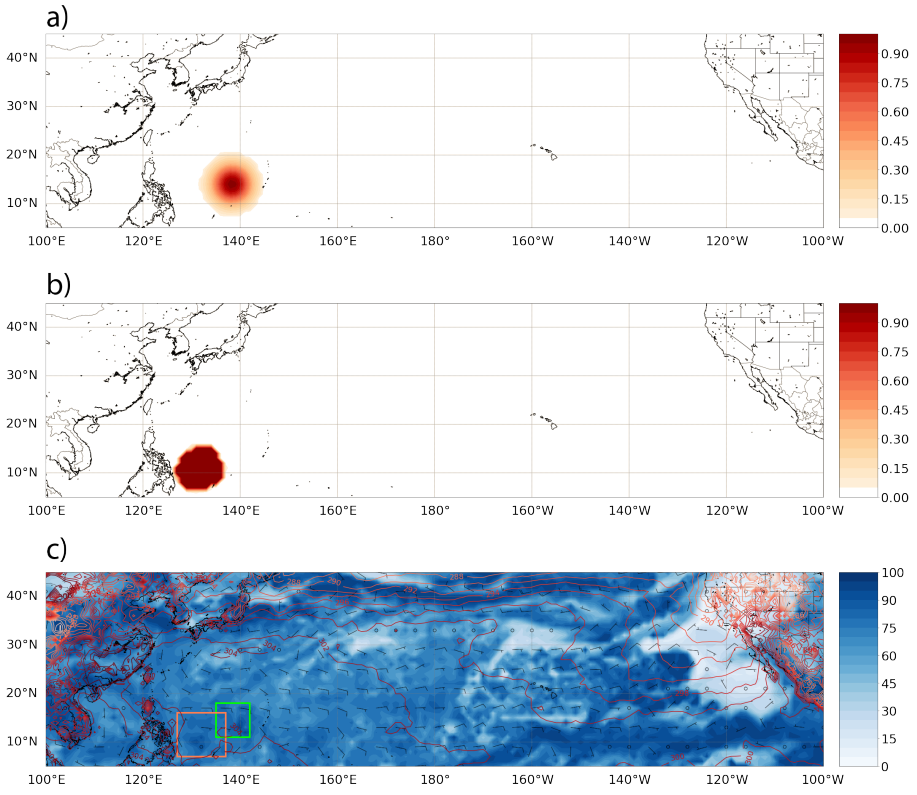

FIG. 7. An example of a true positive TCG prediction case obtained from UNet for Typhoon Chanthu (2021) valid at 0600 UTC Sep 6 that shows a) the observed location of the TCG event (shaded), b) the UNet's prediction of the TCG probability distribution (shaded), and c) the corresponding large-scale environment including relative humidity (shaded, unit %), surface temperature (contours, K), and the surface wind barbs at 850mb. The green box in (c) denotes the observed genesis location of Chanthu

capture this TCG event, thus resulting in a lower performance overall as compared to ResNet when using the F1 score metric as seen in Fig. 6.

Along with the degradation of the UNet performance when we attempt to extract more information on TCG location, note that UNet has the same sensitivity to different input domain sizes as ResNet. Our IG analysis for UNet captures a similar effect of far-field information that is fed into UNet when predicting TCG with the large domain (not shown). That is, a larger domain could allow for more remote information and help improve TCG prediction as compared to a smaller domain. This behavior iterates that far-field environmental information is of significant importance for TCG prediction with DL models, albeit the physical reasons for such a remote contribution of the environment are still elusive. Note again that our IG analyses for UNet also do not answer the

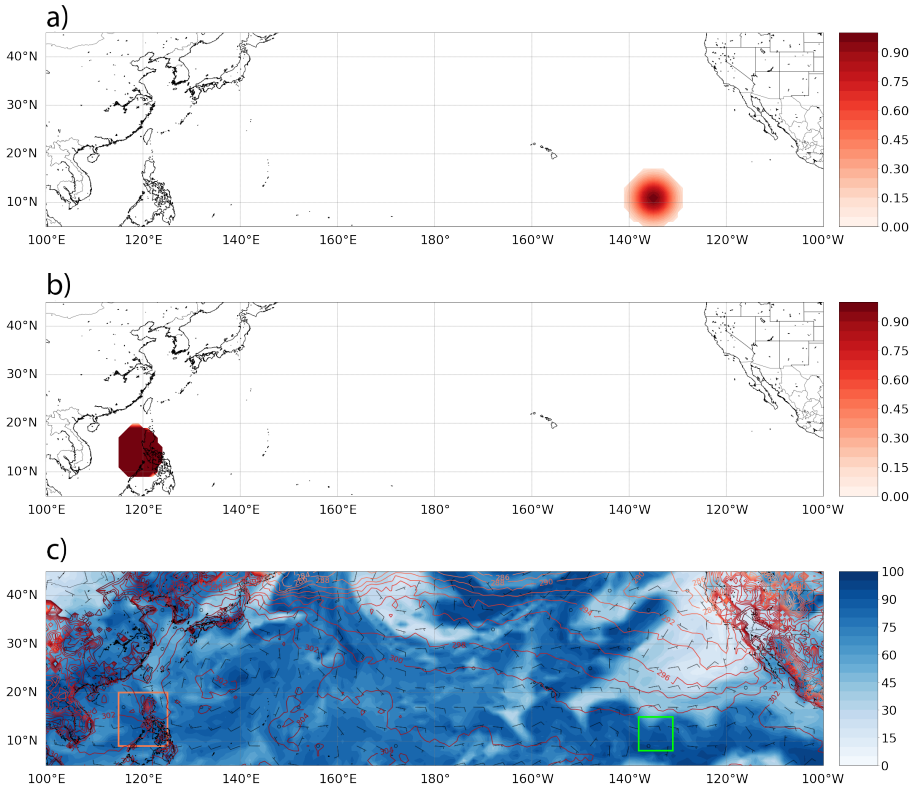

FIG. 8. Similar to Fig. 7 but for a false positive TCG prediction case for Storm 01E (2018) valid at 1200 UTC May 10.

question of what far-field features are most critical for the performance of the ResNet or UNet model, other than the fact that a smaller domain could not contain some far-field information important for TCG prediction. As a result, a larger domain size is essential for better TCG prediction as shown in Figs. 4 and 6.

c. Sensitivity experiments

Because the results for ResNet and UNet shown in the previous section are obtained from one specific model design and hyperparameters, it is of interest to examine next how sensitive these models are to different hyperparameter values. With current ML tools, these sensitivity analyses are generally not necessary in practical implementation as they can be bypassed by using automatic search space. From the research standpoint, understanding how DL models change with different

hyperparameters is however important so one can learn which parameters are the key to the current problem.

In this section, we will present sensitivity analyses for two common hyperparameters in ML models including kernel size and the number of convolutional filters. Other sensitivities such as dropout, strike, or initialization weights are less significant for our problems and so will not be discussed herein. In addition, because ResNet outperforms UNet in terms of TCG detection F1 score, we also limit our sensitivity analyses in this section to the ResNet architecture only. Similar analyses for UNet can be readily carried out, using the same approach and so will not be presented further.

Recall from Fig. 1 that ResNet's architecture is comprised of multiple convolution blocks with a default kernel size 3×3. To see how ResNet depends on the choice of kernel size, we replace the default 3×3 convolution kernel with 5×5 and 7×7. The resulting model is then trained only with the large domain covering the Pacific Ocean, as the small domain does not perform well as shown in the previous section. Figure 9 shows the results from these kernel size experiments. One notices that in general the 5×5 kernel performs better than either 3×3 or 7×7 kernel. For this 5×5 kernel, the model achieves a better precision score, thus increasing the overall F1 score for the available test data. It appears that larger kernels lead to a larger receptive field, thus allowing DL models to get more information from the surrounding area to predict TC formation. However, if the receptive field becomes too large, then the signal-to-noise ratio will decrease and reduce the performance of the model. As a result, the 5×5 kernel performs best in our ResNet model as seen in Fig. 9.

Of course, the best performance of ResNet for the specific kernel size of $5 times 5$ is alone insufficient to generalize for the entire TCG prediction system, as it also depends on many other parameters such as the data sample size, the input domain, the number of channels, etc. Any change in these parameters could alter this sensitivity easily, and so the default kernel of 3×3 is still used in our control design to maintain the performance stability, and computational cost after extensive tests and validations. However, this kernel size sensitivity analysis could at least show that proper tuning of DL hyperparameters is important before one can tailor a DL model for practical applications.

Regarding the sensitivity of ResNet to the number of filters in each convolutional block, Fig. 10 shows the results for experiments in which the first convolutional block has more filters instead 64

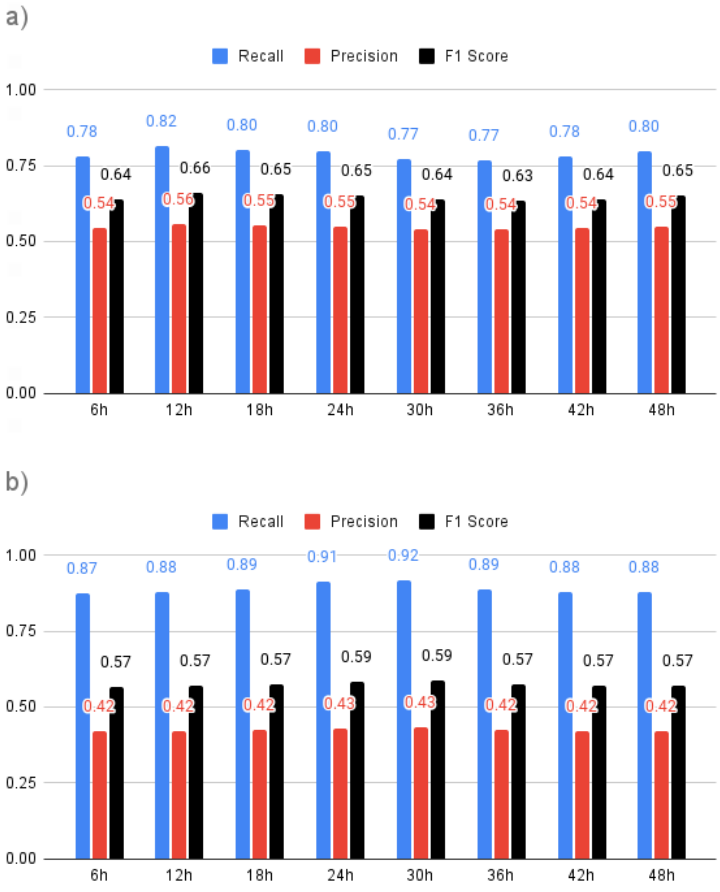

FIG. 9. Similar to Fig. 4 but for ResNet's performance with two different CNN kernel sizes: a) a 5×5, and b) 7×7.

as in the original design, with the next block doubling the number of filters of the preceding one. As seen from Fig. 10, the model with the starting convolutional block of 128 filters performs the best, achieving the highest F1 score of 0.66 at 12-h lead time. This is somewhat expected because ResNet has more capacity to store and learn information about the large-scale environment required for TCG prediction with more filters. However, when the number of filters increases by more than 256, the performance of ResNet starts to decrease, suggesting that more weights also make the model more prone to overfitting, given the same input data that we have. This potential overfitting explains the degradation of ResNet when the number of filters in the first layer is more than 256 as shown in Fig. 10. One can improve this by adding more training data, which is a trade-off that we have to make here due to our limited data record and computational resources.

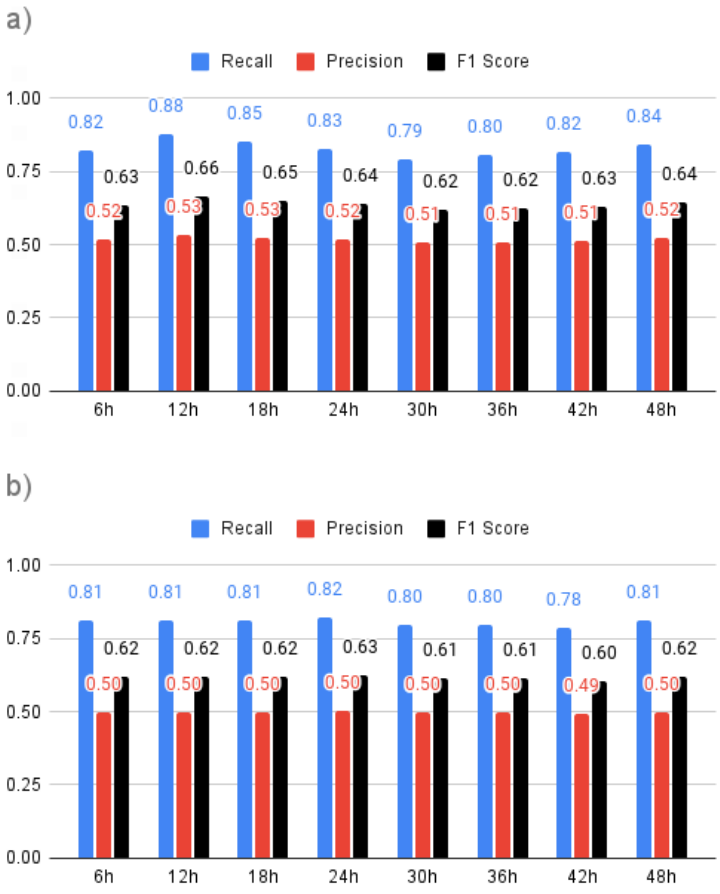

FIG. 10. Similar to Fig. 4 but for ResRet's performance with different numbers of filters in the first convolutional block for a) 128 filters, and b) 256 filters.

Our similar sensitivity analyses for ResNet and UNet using a smaller subdomain in the WP basin confirm that both models tend to perform worse with a smaller domain size for all ranges of kernel sizes and the number of filters. This persistent difference between the large and small input domains reiterates our previous speculation on the contributing far-field environmental factors to the different performances of our DL models. That is, the large-scale environmental factors that govern TCG processes can be better captured in the DL models by picking up potential far-field features in the large domain, which is absent in the small domain. Note also that a larger domain size will generally have more TCG events such that the number of positive cases is larger, thus allowing the models a better chance to learn the correct environmental conditions needed for TCG. Which environmental factors play the most dominant role in our models are, however, unclear from

the above domain size or kernel sensitivity, which require additional analyses that we turn into next.

4. Selection of environment features

From the scientific perspective, determining which environmental factors among the input channels play the most significant role in TCG prediction is important to address. While traditional diagnostic and observational analyses have captured a number of favorable conditions for TCG including warm SST, low shear, high vorticity area, and moisture environment (see, e.g. Gray 1998; McBride and Zehr 1981; Kieu and Zhang 2010, 2009; Halperin et al. 2013; Wang et al. 2019; Vu et al. 2021), being able to further quantify additional factors along with their relative impacts is an advantage of the DL techniques that we wish to present in this section. Unlike the hyperparameter tuning for DL models, feature selection is a different part of DL that can help reveal more physical insights than simply running DL models as a black box. As a part of feature engineering, feature selection is to some extent very similar to the predictor selection processes in traditional statistical research, as it is a way to choose the best possible predictors in a regression model.

There are various ways to do feature selection for DL models. In this study, we apply the forward-selection algorithm that is based on the information gained in filter methods. The algorithm starts first with a list of features that we want to select. It then iterates through the list of features and selects one feature that achieves the best F1 score (or any validation metric) among all. This feature is appended to the list of best-selected features, and the algorithm is then repeated to choose the next best feature until it reaches the number of a desired threshold (see 1). This approach is very close to the Fisher score method that is widely used in supervised feature selection methods by which the resulting outcome returns the ranks of all features based on the Fisher score in descending order. Because UNet does not perform well with our current settings, we will apply the feature selection only for ResNet in this section. The same procedure can be applied for UNet or any DL model, so long as the model performs sufficiently well to allow for adding or removing different features effectively.

Figure 11a-c shows the performance of ResNet with the top 3, 4, and 5 features, which are obtained from the list of 13 input channels using our forward-selection algorithm. These top five features, ranked from the highest to the lowest, are CAPE, horizontal wind components (u and v)

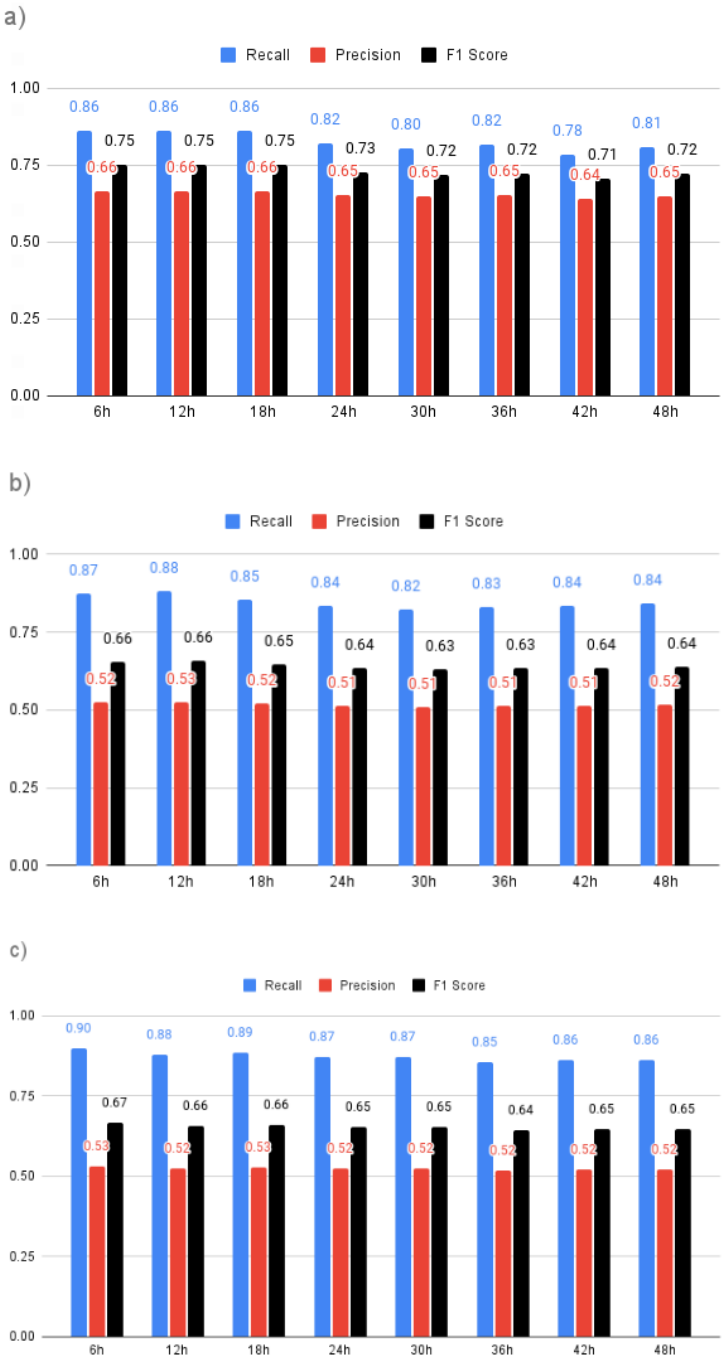

FIG. 11. Similar to Fig. 4 but for ResNet's performance with different dominant features obtained from the forward-feature selection procedure, with the highest ranked features including a) 3 features; b) 4 features; and c) 5 features.

at 850 hPa, horizontal winds at 200 hPa, and sea surface temperature, respectively. One notices from Fig. 11a that ResNet could achieve good performance with just the first 3 features including CAPE and horizontal winds at 850 hPa. Adding horizontal wind at 200 hPa however results in a drop in the overall performance as seen in Fig. 11b, yet including the fifth feature (i.e., SST) could lead to an overall increased performance similar to using all 13 features. This intriguing behavior confirms that important features for TCG prediction do not add up linearly, but they have to go in a group to best characterize TCG processes. In fact, including more features beyond these above features turns out to be of no further help in terms of the F1 score (not shown).

From the physical standpoint, the above dominant features are somewhat expected and consistent with previous studies on environmental conditions for TCG, using observational analyses and physical-based models. Consider, for example, the 850 hPa-horizontal winds captured in the top three features. Essentially, these features represent the low-level vorticity, whose importance is consistent with the previous finding about the requirement of a pre-existing tropical disturbance for TCG (see, e.g., Gray 1982; Nolan et al. 2007; Kieu and Zhang 2009). Likewise, the CAPE and SST features capture the maximum potential intensity limit, which has been also known to be vital and included in the genesis potential index (e.g., Emanuel and Nolan 2004b; Nolan et al. 2007; Camargo et al. 2014; Vu et al. 2021; Tang et al. 2020; Kieu et al. 2023). The environmental shear factor is also captured by our feature-selection analyses, with the 200-hPa zonal wind feature selected in the top five features. In this regard, the feature-selection analyses could confirm the previous findings on the required conditions for TCG, while at the same time revealing some intriguing behaviors when different features must go in a group in the DL models beyond the traditional genesis index.

It should be mentioned that the findings on the dominant large-scale factors for TCG obtained herein are very specific to the ResNet architecture, and they may change with different settings, hyperparameters, kernel sizes, or input data length. Nonetheless, the approach and the potential implication of these results are still significant, as they suggest that ML algorithms can be customized for TCG prediction when more training data is available. In particular, our approach presents a way that one can refine and obtain a new understanding of TCG processes beyond the traditional way of using numerical sensitivity experiments, so long as our computational efficiency can be improved to process longer global data.

Algorithm 1 Forward Feature Selection Algorithm

```
procedure FORWARD SELECTION(features, nbFeatures)
    nbChosenFeatures ← 0
    chosenFeatures ← []
    while nbChosenFeatures < nbFeatures do
        bestAccuracy ← 0.0
        remainingFeatures ← features not in chosenFeatures
        for f ∈ remainingFeatures do
            featuresToUse ← chosenFeatures + f
            model ← train model with featuresToUse
            accuracy ← evaluate model
            if accuracy > bestAccuracy then
                bestAccuracy ← accuracy
                bestFeatures ← chosenFeatures + f
            end if
        end for
        chosenFeatures ← bestFeatures
        nbChosenFeatures ← nbChosenFeatures + 1
    end while
    return chosenFeatures
end procedure
```

5. Conclusion

In this study, the potential applicability of deep learning models for tropical cyclogenesis (TCG) prediction was examined. Unlike the typical classification problems that focus on answering a binary question of yes or no from existing features, TCG prediction is unique because there exists no clear TC circulation or characteristics from input data at the time one wants to predict a TCG event. Predicting TCG at different forecast lead times would therefore require a different design such that information on a TCG event can be detected even before the emergence of any TC signal for practical purposes.

Specifically in this study, two popular DL architectures including ResNet and UNet were used to examine the capabilities of convolutional neural networks for TCG prediction. These architectures are to some extent complementary to each other, as ResNet can provide yes/no prediction for a TCG event while UNet could provide additional information on the location of the TCG event. With a hypothesis that TCG must require some specific conditions detectable from the large-scale environment, we extracted from the NCEP/NCAR reanalysis dataset a set of meteorological fields (features) that are known to be most critical for TCG from previous studies. These fields were then treated as input channels of an image for our DL models. Using the best track data to label

TCG events at different forecast lead times, we could train our DL models and obtain a number of significant results relating to their capability in TCG prediction for practical applications.

First, applying ResNet and UNet to predict TCG for an illustrative period from 2005-2020 showed that both models are capable of predicting TCG with the F1 score ranging from 0.25-0.63. Of interest, the F1 score in both models shows a maximum value at 18-36 hour lead time, and gradually decreases at longer lead times. Such decaying performance with forecast lead times in both DL models is a noteworthy result, given that any prediction from these models is based purely on a given state without any physical principles or dynamical equations as in numerical weather prediction models. We wish to emphasize herein the predictability implication of our result, as our approach does actually predict TCG from a given initial field. This is very different from applying ML models on a global model forecast, which is basically an ML downscaling (or detection) of the gridded forecast field and so it possesses little predictability implication. The fact that both ResNet and UNet could capture decaying predictability with forecast lead time as obtained in this study suggests that these DL algorithms are able to capture the expected evolution of the atmosphere, even without any governing equations.

Second, our analyses of the ResNet and UNet performance for two different input data sizes including 1) a large domain covering most of the North Pacific Ocean and 2) a small subdomain covering a part of the northwestern Pacific basin showed that the use of a large domain gives overall better TCG prediction. Specifically, the F1 score for the large domain input is about 40% higher than that obtained from the smaller domain at all forecast lead times. Using the integrated gradient analyses, it was found that the large domain could take into account some far-field information, which helps improve the prediction of TCG overall. In addition, the use of the large domain also allows for more TCG labels, which reduces the data unbalance issue and results in better performance. This is another significant finding, because it reveals the sensitivity of machine learning to the data domain in TCG prediction. While machine learning algorithms do not require any dynamic constraints *a priori*, they do need to access information from different places in the domain to correctly detect favorable conditions for TCG. As such, a proper choice of input data size is critical for the TCG prediction application.

Additional sensitivity experiments with different hyperparameters showed that the kernel size appears to be more important than the number of filters or the number of conventional blocks in

ResNet. In fact, ResNet reaches its peak performance with a kernel size of 5×5 and 128 filters. A larger kernel size or more filters would not help improve the performance of ResNet further. Between ResNet and UNet, we also found that the performance of ResNet is overall much higher than the UNet in predicting TCG for all ranges of hyperparameters and lead times. Specific to the data and architectures used in this study, ResNet's F1 score is on average almost 2 times that obtained from UNet. This is expected because UNet provides not only the probability distribution but also the location of TCG events. The more information one wishes to extract from a DL model, the more likely the model would make errors and so become less accurate.

By further applying the feature selection method for different data input channels, we could confirm several important environmental factors for TCG prediction in the Pacific Ocean, which includes CAPE, horizontal wind components (u and v) at 850 hPa and 200 hPa, and sea surface temperature. These factors are consistent with the well-known TCG requirements obtained from the previous modeling and observational studies. The advantage of our DL approach is that additional features could be searched and ranked for different basins and forecast lead times when a DL model is fully optimized and more data is used. In this work, both of our DL models are of course still underperform due to the limit in computational resources and input data, which prevents us from carrying out full feature selection analyses. Further examination and tuning of different DL architectures, including the possible use of recurrent neural networks to take into account the temporal component of data, are currently under development for which we will update in our upcoming studies.

Acknowledgments. This research was partially supported by the National Science Foundation (NSF Award AGS-2309929). We thank two anonymous reviewers for their constructive comments and suggestions, which have helped improve this work significantly.

Data availability statement. The NCEP/NCAR FNL dataset used in this study is available at National Centers for Environmental Prediction, National Weather Service, NOAA, U.S. Department of Commerce (2000), and the best track TC data is available at https://www.ncei.noaa.gov/products/international-best-track-archive. The ResNet and UNet models used in this study can be freely accessed through our Github repository at https://github.com/kieucq/deep_learning_tc_prediction.

References

Abadi, M., and Coauthors, 2015: TensorFlow: Large-scale machine learning on heterogeneous systems. URL https://www.tensorflow.org/, software available from tensorflow.org.

Camargo, S. J., M. K. Tippett, A. H. Sobel, G. A. Vecchi, and M. Zhao, 2014: Testing the performance of tropical cyclone genesis indices in future climates using the hiram model. *Journal of Climate*, **27 (24)**, 9171–9196.

Chen, T.-C., M.-C. Yen, J.-D. Tsay, J. Alpert, and N. T. Tan Thanh, 2012: Forecast advisory for the late fall heavy rainfall/flood event in central vietnam developed from diagnostic analysis. *Weather and forecasting*, **27 (5)**, 1155–1177.

Cossuth, J. H., R. D. Knabb, D. P. Brown, and R. E. Hart, 2013: Tropical cyclone formation guidance using pregenesis dvorak climatology. part i: Operational forecasting and predictive potential. *Weather and forecasting*, **28 (1)**, 100–118.

DeMaria, M., J. A. Knaff, and B. H. Connell, 2001: A tropical cyclone genesis parameter for the tropical atlantic. *Weather and Forecasting*, **16 (2)**, 219–233.

Emanuel, K., and D. S. Nolan, 2004a: Tropical cyclone activity and the global climate system. *26th conference on hurricanes and tropical meteorolgy*.

Emanuel, K. A., and D. S. Nolan, 2004b: Tropical cyclone activity and the global climate system. *26th Conference on Hurricanes and Tropical Meteorology*, **10A.2**.

Fenner, M., 2019: *Machine learning with Python for everyone*. Addison-Wesley Professional.

Ferrara, M., F. Groff, Z. Moon, K. Keshavamurthy, S. M. Robeson, and C. Kieu, 2017: Large-scale control of the lower stratosphere on variability of tropical cyclone intensity. *Geophysical Research Letters*, **44 (9)**, 4313–4323, doi:https://doi.org/10.1002/2017GL073327.

Gao, S., P. Zhao, B. Pan, Y. Li, M. Zhou, J. Xu, S. Zhong, and Z. Shi, 2018: A nowcasting model for the prediction of typhoon tracks based on a long short term memory neural network. *Acta Oceanologica Sinica*, **37 (5)**, 8–12.

Giffard-Roisin, S., M. Yang, G. Charpiat, C. Kumler Bonfanti, B. Kégl, and C. Monteleoni, 2020: Tropical cyclone track forecasting using fused deep learning from aligned reanalysis data. *Frontiers in big Data*, **3**, 1.

Goodfellow, I., Y. Bengio, and A. Courville, 2017: Deep learning (adaptive computation and machine learning series). *Cambridge Massachusetts*, 321–359.

Goswami, P., and G. Mohapatra, 2014: A comparative evaluation of impact of domain size and parameterization scheme on simulation of tropical cyclones in the bay of bengal. *Journal of Geophysical Research: Atmospheres*, **119 (1)**, 10–22.

Gray, W. M., 1982: Tropical cyclone genesis and intensification: Intense atmospheric vortices. *Topics in Atmospheric and Oceanographic Sciences*, **10**, 3–20.

Gray, W. M., 1998: The formation of tropical cyclones. *Meteorology and atmospheric physics*, **67 (1)**, 37–69.

Halperin, D. J., H. E. Fuelberg, R. E. Hart, and J. H. Cossuth, 2016: Verification of tropical cyclone genesis forecasts from global numerical models: Comparisons between the north atlantic and eastern north pacific basins. *Weather and Forecasting*, **31 (3)**, 947 – 955, doi:https://doi.org/10.1175/WAF-D-15-0157.1.

Halperin, D. J., H. E. Fuelberg, R. E. Hart, J. H. Cossuth, P. Sura, and R. J. Pasch, 2013: An evaluation of tropical cyclone genesis forecasts from global numerical models. *Weather and Forecasting*, **28 (6)**, 1423–1445.

Halperin, D. J., A. B. Penny, and R. E. Hart, 2020: A comparison of tropical cyclone genesis forecast verification from three global forecast system (gfs) operational configurations. *Weather and Forecasting*, **35 (5)**, 1801 – 1815, doi:https://doi.org/10.1175/WAF-D-20-0043.1.

Hastie, T., R. Tibshirani, J. H. Friedman, and J. H. Friedman, 2009: *The elements of statistical learning: data mining, inference, and prediction*, Vol. 2. Springer.

He, K., X. Zhang, S. Ren, and J. Sun, 2015: Deep residual learning for image recognition. *arXiv*, URL http://arxiv.org/abs/1512.03385.

Henderson, S. A., and E. D. Maloney, 2013: An intraseasonal prediction model of atlantic and east pacific tropical cyclone genesis. *Monthly Weather Review*, **141 (6)**, 1925–1942.

Hennon, C. C., C. N. Helms, K. R. Knapp, and A. R. Bowen, 2011: An objective algorithm for detecting and tracking tropical cloud clusters: Implications for tropical cyclogenesis prediction. *Journal of Atmospheric and Oceanic Technology*, **28 (8)**, 1007 – 1018, doi:10.1175/2010JTECHA1522.1.

Hennon, C. C., and Coauthors, 2013: Tropical cloud cluster climatology, variability, and genesis productivity. *Journal of Climate*, **26 (10)**, 3046 – 3066, doi:10.1175/JCLI-D-12-00387.1.

Hill, K. A., and G. M. Lackmann, 2011: The impact of future climate change on tc intensity and structure: A downscaling approach. *Journal of Climate*, **24 (17)**, 4644–4661, doi:10.1175/2011JCLI3761.1.

Houze, R. A., 1982: Cloud clusters and large-scale vertical motions in the tropics. *J. Meteor. Soc. Japan*, **60**, 396–410.

Karyampudi, V. M., and H. F. Pierce, 2002: Synoptic-scale influence of the saharan air layer on tropical cyclogenesis over the eastern atlantic. *Monthly weather review*, **130 (12)**, 3100–3128.

Kieu, C., and D.-L. Zhang, 2018: The control of environmental stratification on the hurricane maximum potential intensity. *Geophysical Research Letters*, **45 (12)**, 6272–6280, doi:https://doi.org/10.1029/2018GL078070.

Kieu, C., M. Zhao, Z. Tan, B. Zhang, and T. Knutson, 2023: On the role of sea surface temperature in the clustering of global tropical cyclone formation. *Journal of Climate*, 1 – 39, doi:10.1175/JCLI-D-22-0623.1.

Kieu, C. Q., and D.-L. Zhang, 2008: Genesis of tropical storm eugene (2005) from merging vortices associated with itcz breakdowns. part i: Observational and modeling analyses. *Journal of the Atmospheric Sciences*, **65 (11)**, 3419 – 3439, doi:10.1175/2008JAS2605.1.

Kieu, C. Q., and D.-L. Zhang, 2009: Genesis of tropical storm eugene (2005) from merging vortices associated with itcz breakdowns. part ii: Roles of vortex merger and ambient potential vorticity. *Journal of the Atmospheric Sciences*, **66 (7)**, 1980 – 1996, doi:10.1175/2008JAS2905.1.

Kieu, C. Q., and D.-L. Zhang, 2010: Genesis of tropical storm eugene (2005) from merging vortices associated with itcz breakdowns. part iii: Sensitivity to various genesis parameters. *Journal of the Atmospheric Sciences*, **67 (6)**, 1745 – 1758, doi:10.1175/2010JAS3227.1.

Kim, M., M.-S. Park, J. Im, S. Park, and M.-I. Lee, 2019: Machine learning approaches for detecting tropical cyclone formation using satellite data. *Remote Sensing*, **11 (10)**, doi:10.3390/rs11101195, URL https://www.mdpi.com/2072-4292/11/10/1195.

Kingma, D. P., and J. Ba, 2014: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Knapp, K. R., M. C. Kruk, D. H. Levinson, H. J. Diamond, and C. J. Neumann, 2010: The international best track archive for climate stewardship (ibtracs). *Bull. Amer. Meteor. Soc.*, **91**, 363–376.

Li, W., Z. Wang, and M. S. Peng, 2016: Evaluating tropical cyclone forecasts from the ncep global ensemble forecasting system (gefs) reforecast version 2. *Weather and Forecasting*, **31 (3)**, 895–916.

Lin, T.-Y., P. Goyal, R. Girshick, K. He, and P. Dollár, 2017: Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Matsuoka, D., M. Nakano, D. Sugiyama, and S. Uchida, 2018: Deep learning approach for detecting tropical cyclones and their precursors in the simulation by a cloud-resolving global nonhydrostatic atmospheric model. *Progress in Earth and Planetary Science*, doi:10.1186/s40645-018-0245-y, URL https://doi.org/10.1186/s40645-018-0245-y.

McBride, J. L., and R. Zehr, 1981: Observational analysis of tropical cyclone formation. part ii: Comparison of non-developing versus developing systems. *Journal of the Atmospheric Sciences*, **38 (6)**, 1132–1151.

Miller, J., M. Maskey, and T. Berendes, 2017: Using deep learning for tropical cyclone intensity estimation. *AGU Fall Meeting Abstracts*, Vol. 2017, IN11E–05.

Murphy, K. P., 2012: *Machine learning: a probabilistic perspective*. MIT press.

National Centers for Environmental Prediction, National Weather Service, NOAA, U.S. Department of Commerce, 2000: Ncep fnl operational model global tropospheric analyses, continuing from july 1999. Research Data Archive at the National Center for Atmospheric Research, Computational and Information Systems Laboratory, Boulder CO, URL https://doi.org/10.5065/D6M043C6.

Nguyen, Q., 2023: Deep learning for tropical cyclone formation detection. ProQuest Dissertations Publishing, Indiana University, 120p.

Nolan, D., E. D. Rappin, and K. A. Emanuel, 2007: Tropical cyclogenesis sensitivity to environmental parameters in radiative–convective equilibrium. *Quart. J. Roy. Meteor. Soc.*, **133**, 2085–2107.

Park, M.-S., M. Kim, M.-I. Lee, J. Im, and S. Park, 2016: Detection of tropical cyclone genesis via quantitative satellite ocean surface wind pattern and intensity analyses using decision trees. *Remote sensing of environment*, **183**, 205–214.

Peng, M. S., B. Fu, T. Li, and D. E. Stevens, 2012: Developing versus nondeveloping disturbances for tropical cyclone formation. part i: North atlantic. *Monthly weather review*, **140 (4)**, 1047–1066.

Ronneberger, O., P. Fischer, and T. Brox, 2015: 2015-u-net. *arXiv*, 1–8, URL http://lmb.informatik.uni-freiburg.de/%0Aarxiv:1505.04597v1.

Su, H., L. Wu, J. H. Jiang, R. Pai, A. Liu, A. J. Zhai, P. Tavallali, and M. DeMaria, 2020: Applying satellite observations of tropical cyclone internal structures to rapid intensification forecast with machine learning. *Geophysical Research Letters*, **47 (17)**, e2020GL089 102.

Sundararajan, M., A. Taly, and Q. Yan, 2017: Axiomatic attribution for deep networks. *34th International Conference on Machine Learning, ICML 2017*, **7**, 5109–5118, URL https://arxiv.org/abs/1703.01365v2.

Tang, B. H., and Coauthors, 2020: Recent advances in research on tropical cyclogenesis. *Tropical Cyclone Research and Review*, **9 (2)**, 87–105.

Tien, T. T., D. N.-Q. Hoa, C. Thanh, and C. Kieu, 2020: Assessing the impacts of augmented observations on the forecast of typhoon wutip's (2013) formation using the ensemble kalman filter. *Weather and Forecasting*, **35 (4)**, 1483 – 1503, doi:10.1175/WAF-D-20-0001.1.

Vu, T.-A., C. Kieu, D. Chavas, and Q. Wang, 2021: A numerical study of the global formation of tropical cyclones. *Journal of Advances in Modeling Earth Systems*, **13 (1)**, e2020MS002 207, doi:https://doi.org/10.1029/2020MS002207.

Wang, Q., C. Kieu, and T.-A. Vu, 2019: Large-scale dynamics of tropical cyclone formation associated with itcz breakdown. *Atmospheric Chemistry and Physics*, **19 (13)**, 8383–8397, doi:10.5194/acp-19-8383-2019, URL https://www.atmos-chem-phys.net/19/8383/2019/.

Yamaguchi, M., and N. Koide, 2017: Tropical cyclone genesis guidance using the early stage dvorak analysis and global ensembles. *Weather and Forecasting*, **32 (6)**, 2133 – 2141, doi: https://doi.org/10.1175/WAF-D-17-0056.1.

Zhang, T., W. Lin, Y. Lin, M. Zhang, H. Yu, K. Cao, and W. Xue, 2019: Prediction of tropical cyclone genesis from mesoscale convective systems using machine learning. *Weather and Forecasting*, **34**, 1035–1049, doi:10.1175/WAF-D-18-0201.1, URL https://journals.ametsoc.org/doi/10.1175/WAF-D-18-0201.1.

Zhang, W., B. Fu, M. S. Peng, and T. Li, 2015: Discriminating developing versus nondeveloping tropical disturbances in the western north pacific through decision tree analysis. *Weather and Forecasting*, **30 (2)**, 446–454.