2019 IEEE Winter Conference on Applications of Computer Vision

Deep-Hurricane-Tracker: Tracking and Forecasting Extreme Climate Events

Sookyung Kim[1], Hyojin Kim[1], Joonseok Lee[2], Sangwoong Yoon[3],
Samira E. Kahou[4], Karthik Kashinath[5], Mr Prabhat[5]
Lawrence Livermore Nat'l Lab.[1] Google Research[2] Seoul Nat'l Univ.[3]
Microsoft Research[4] Lawrence Berkeley Nat'l Lab.[5]
{kim79,kim63}@llnl.gov, joonseok@google.com, sangwoong24yoon@gmail.com,
samira.ebrahimi.kahou@gmail.com, {prabhat,kkashinath}@lbl.gov

Abstract

Tracking and predicting extreme events in large-scale spatio-temporal climate data are long standing challenges in climate science. In this paper, we propose Convolutional LSTM (ConvLSTM)-based spatio-temporal models to track and predict hurricane trajectories from large-scale climate data; namely, pixel-level spatio-temporal history of tropical cyclones. To address the tracking problem, we model time-sequential density maps of hurricane trajectories, enabling to capture not only the temporal dynamics but also spatial distribution of the trajectories. Furthermore, we introduce a new trajectory prediction approach as a problem of sequential forecasting from past to future hurricane density map sequences. Extensive experiment on actual 20 years record shows that our ConvLSTM-based tracking model significantly outperforms existing approaches, and that the proposed forecasting model achieves successful mapping from predicted density map to ground truth.

1. Introduction

Tracking and predicting extreme climate events are pressing and challenging problems that humanity has faced and tackled for a long time. Traditionally, human experts detected extreme climate events based on hand-picked thresholds [9] based on high-resolution spatio-temporal climate data from physics-based simulations. Most conventional methods for extreme climate event detection have been built upon human expertise based on related spatio-temporal physics variables with manually-picked, often subjective, thresholds. For example, hurricane (tropical cyclone especially in the Caribbean) is generally defined as a large rotating storm with high speed winds over warm water in tropical areas with low pressure at the center. However, there is no strictly defined criteria for those variables; for instance, there exists different interpretations to define how “low” the center pressure is, or how “warm” the temperature is among climate scientists. Therefore, different thresholds are used to characterize hurricanes.

Recent advances in machine learning provide efficient solutions for object detection and tracking problems, which potentially overcome issues with conventional labor-intensive and subjective extreme climate event detection. Deep neural networks, for example, learn complex patterns between variables from data samples, potentially avoiding conventional threshold-based criteria of climate variables for event detection.

As a special case of object detection and tracking problems, there are some unique properties of extreme climate events tracking, distinguishing it from other object recognition problems:

1. The target event is dependent on complex spatio-temporal dynamics between multiple scientific variables. Particularly, the target is affected by **wide-range of spatial dynamics**, known as the “butterfly” effect.
2. Climate events span for a long period of time, meaning that we need to effectively model **long-term temporal dependencies** between variables and the target.
3. The **number of target objects** in each time frame can be **arbitrary**, from none to multiples. It is therefore not easy to pre-define the number of target objects to detect and track.

Over the past few years, researchers in climate informatics have made significant progress in developing models to solve the problem of extreme climate event detection by applying various pattern recognition techniques such as neural networks. However, their models are limited in that they do not fully take the properties above into account. Several works, for example, proposed fully supervised convolutional neural networks (CNNs) to detect and classify well-known types of extreme climate events with high precision. [7, 6] However, those models do not take temporal dynamics of extreme climate events into account (property 2 above), which is an essential component in defining

978-1-7281-1975-5/19/$31.00 ©2019 IEEE
DOI 10.1109/WACV.2019.00192

them. Models based on single shot might mislead into incorrect predictions, due to the lack of essential temporal dynamics of real hurricanes. Also, most existing CNN-based classification models for extreme climate events are trained on cropped patches from a small region, not on the entire global map, making it difficult to localize the events in global map, failing to tackle property 1.

A recent work by Racah et al. [15] proposed a spatio-temporal auto-encoding CNNs, considering temporal aspects for extreme climate event detection. However, their design of encoder-decoder architecture with concatenated time axis as the third dimension limits the model to take only a few time steps as input, failing to fully address the property 2 above. This architecture does not allow arbitrarily long input sequences as the input size linearly grows with the sequence length. Therefore, a recursive model such as GRU or LSTM with memorizing mechanism is required to leverage long-term temporal information.

In this work, we tackle the extreme climate event detection by formulating the problem as a regression problem, learning a mapping from time series climate variables to time series density map of event probabilities. By modeling the output as a density map, the proposed model is capable of detecting arbitrary number of extreme climate events, instead of setting it as a hyper-parameter (property 3). Particularly, we propose a variant of Convolutional LSTM (ConvLSTM) model, which effectively captures pixel-level latent spatio-temporal patterns existing in multivariate climate variables. The *Conv* part extracts spatial patterns, while the *LSTM* part captures the temporally evolving patterns. Therefore, the proposed model resolves stated problems above to detect target objects (e.g, hurricanes) in large spatial region (property 1) considering long-term temporal dynamics (property 2). We verify superior performance of the proposed approach by comparing against a couple of state-of-the-art baselines among extreme climate detection models using deep neural networks: a region-based CNN [7] and 3D convolutional encoder-decoder [15].

Our contribution mainly lies in the unified framework for tracking and forecasting of extreme climate events using ConvLSTM, with the details listed below:

1. We propose an accurate extreme climate event tracking model based on ConvLSTM, simultaneously considering *spatio-temporal correlations* between multiple scientific variables as well as long temporal dynamics in defining events. Our model does not require the number of target objects as a hyper-parameter.

2. We demonstrate *outstanding performance* of our model by comparing detection accuracy against two state-of-the-art baselines in community.

3. In addition to trajectory detection, we present a showcase of proposed model to *predict future trajectories* of hurricane by applying sequence-to-sequence model with ConvLSTM.

The rest of this paper is organized as follows. Section 2 reviews related work in literature. In Section 3–4, we present our proposed ConvLSTM model for extreme climate event tracking and forecasting, respectively, followed by experiments and analysis in Section 5. We conclude with future work in Section 6.

2. Related Work

Toolkit for Extreme Climate Analysis (TECA) [13, 16] is an expert-engineered system with a collection of climate analysis algorithms for extreme event detection, tracking, and other event pattern characterization. Due to the growing popularity of the system and analysis datasets in the climate research community, we use their hurricane dataset as ground truth for our experiment.

Prior to the recent machine learning-driven approaches, a number of numerical model simulation-based methods have been proposed for several decades. To predict and track extreme climate events such as hurricane and other cyclone, most non-machine learning approaches use an ensemble of multiple prediction models or multi-scale prediction systems [24, 3, 19, 12, 11, 20, 22, 14, 21, 2].

The climate research community has recently leveraged the power of machine learning and the availability of large climate datasets. Several neural network-based approaches have been proposed to address problems in climate and weather prediction. In particular, due to the nature of the climate datasets and their inherently temporal analysis, Recurrent Neural Network (RNN)-based algorithms to infer temporal dynamics of climate events have received great attention. Shi et al. [17] introduced the convolutional LSTM (ConvLSTM) architecture for forecasting future precipitation. Trained on two-dimensional radar map time series, their system outperformed the existing precipitation forecasting systems. As the extension of the ConvLSTM, Shi et al. [18] proposed the trajectory GRU that can learn the location-variant structure for recurrent connections.

Liu et al. [7] adopted region-based convolutional architectures (R-CNNs) to predict the class labels for different extreme climate event types, given the climate event dataset with the output of TECA analysis as ground truth. Racah et al. [15] presented a model for extreme climate event detection by using spatio-temporal Convolutional encoder-decoder architecture for bounding box prediction of extreme climate events, along with the ground truth label using TECA analysis. Alemany et al. [1] adopted the fully connected RNN model to predict the trajectory of hurricanes. Kim et al. [5] proposed a prediction method for route trajectories using an incremental neural network.

The ConvLSTM has been also applied to other spatio-temporal learning tasks. Finn et al. [4] demonstrated the

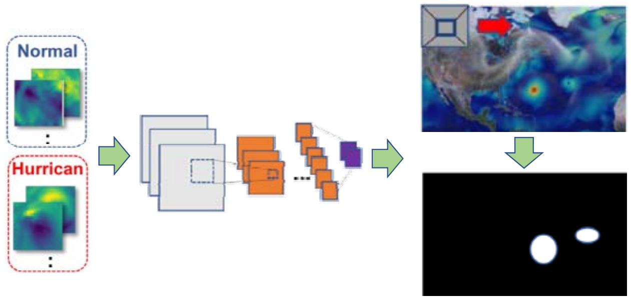
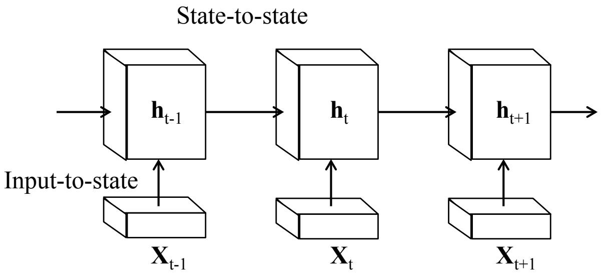


Figure 1. Detection-based CNN tracking model, as a variant of baseline [7].



$$
\begin{aligned}
i_t &= \sigma(W_{xi} * \mathbf{X}_t + W_{hi} * \mathbf{h}_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * \mathbf{X}_t + W_{hf} * \mathbf{h}_{t-1} + W_{cf} \circ C_{t-1} + b_f) \\
C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * \mathbf{X}_t) + W_{hc} * \mathbf{h}_{t-1} + b_c \\
o_t &= \sigma(W_{xo} * \mathbf{X}_t + W_{ho} * \mathbf{h}_{t-1} + W_{co} \circ C_t + b_o) \\
\mathbf{h}_t &= o_t \circ \tanh(C_t)
\end{aligned}
$$

Figure 2. The ConvLSTM Model [17]

ConvLSTM-based motion prediction. Villegas et al. [23] deployed the stacked ConvLSTM for pixel-level prediction in natural videos. Feng et al. [25] applied the ConvLSTM for crowd counting in videos.

3. Tracking Hurricane Trajectories

In this paper, we tackle two related problems: tracking hurricane trajectories and predicting future location of the hurricanes. We present our ConvLSTM-based approach for tracking problem in this section.

3.1. Problem Formulation

The extreme climate event tracking is an example of the multi-object tracking problem, a task of locating multiple objects in a video (or a series of images), maintaining their identities, and yielding their individual trajectories given an input video [10]. Formally, the input tensor $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_T\}$ consists of consecutive T climate images containing hurricane trajectories. There may be multiple trajectories in an input video, and each hurricane may start or end at any frame. Each $\mathbf{X}_i \in \mathbb{R}^{m \times n \times c}$, for $i = 1, 2, ..., T$, is an $m \times n$ 2-D climate image with c climate variables (e.g, wind velocity, precipitation), corresponding to color components (e.g, RGB) in regular images. The label $\mathbf{y}_i$ is a set of k coordinates $\{\mathbf{y}_i^1, \mathbf{y}_i^2, ..., \mathbf{y}_i^k\}$ of hurricane centers existing in the corresponding input image $\mathbf{X}_i$. The number of hurricanes k in ground truth may vary for each input image.

As we do not know the number of objects to be detected in each time frame, it is not straightforward to estimate $\hat{\mathbf{y}}_i$ as a set like $\mathbf{y}_i$. Instead of directly working on the set, we represent both prediction and ground truth as density maps. That is, we estimate a density map $\hat{\mathbf{Y}}_i \in \mathbb{R}^{m \times n}$ with the probability of an hurricane observed in each pixel. Assuming that hurricanes are in circular shape, we represent ground truth labels as a density-map $\mathbf{Y}_i \in \mathbb{R}^{m \times n}$ by Gaussian mixtures centered on each element in $\mathbf{y}_i$. In this setting, we model the tracking problem as a pixel-wise regression problem from $\mathbf{X}$ to $\mathbf{Y}$, minimizing the reconstruction error between the ground truth $\mathbf{Y}$ and our prediction $\hat{\mathbf{Y}}$.

3.2. Baseline Model: Detection-based Tracking

We first review a simpler baseline model, detection-based tracking. As the hurricane detection can be solved using region-based CNNs in high precision (see relevant work in Section 2), we can detect hurricanes in each frame independently, and stitch the results to neighboring frames by connecting with the nearest object among the detected objects in the next time frame. Formally, we independently estimate a density-map $\mathbf{Y}_i$ from $\mathbf{X}_i$ for each $i = 1, ..., T$, ignoring temporal correlation between neighboring images.

Figure 1 shows our detection-based CNN model for hurricane tracking, consisting of two steps. First, we pre-train a convolution-based binary classifier, deciding if an input image patch contains an hurricane or not. The trained CNN model captures the essential characteristics of hurricanes. At the second step, we scan the global map with the trained features to estimate pixel-wise probability of hurricane occurrence. To speed up the convolution operation, we deploy fully convolutional networks (FCN) [8] instead of looping over the entire image with pre-trained feature.

3.3. Our Approach: ConvLSTM-based Model

The tracking-by-detection model in Section 3.2 does not fully take advantage of temporal correlations in the input data sequence. Instead of detecting extreme events independently in each frame, we utilize ConvLSTM as a regression model for mapping from time-series climate images to time-series density-map of hurricane events, effectively exploiting temporal information in addition to spatial relations.

The ConvLSTM model is an extension of the LSTM by replacing fully-connected layers with convolutional structures both in the input-to-state and state-to-state connections to better capture spatial and temporal dependencies of the video, illustrated in Figure 2. Like LSTM, ConvLSTM has a hidden state $\mathbf{h}_t$ per each cell, recurrently con-

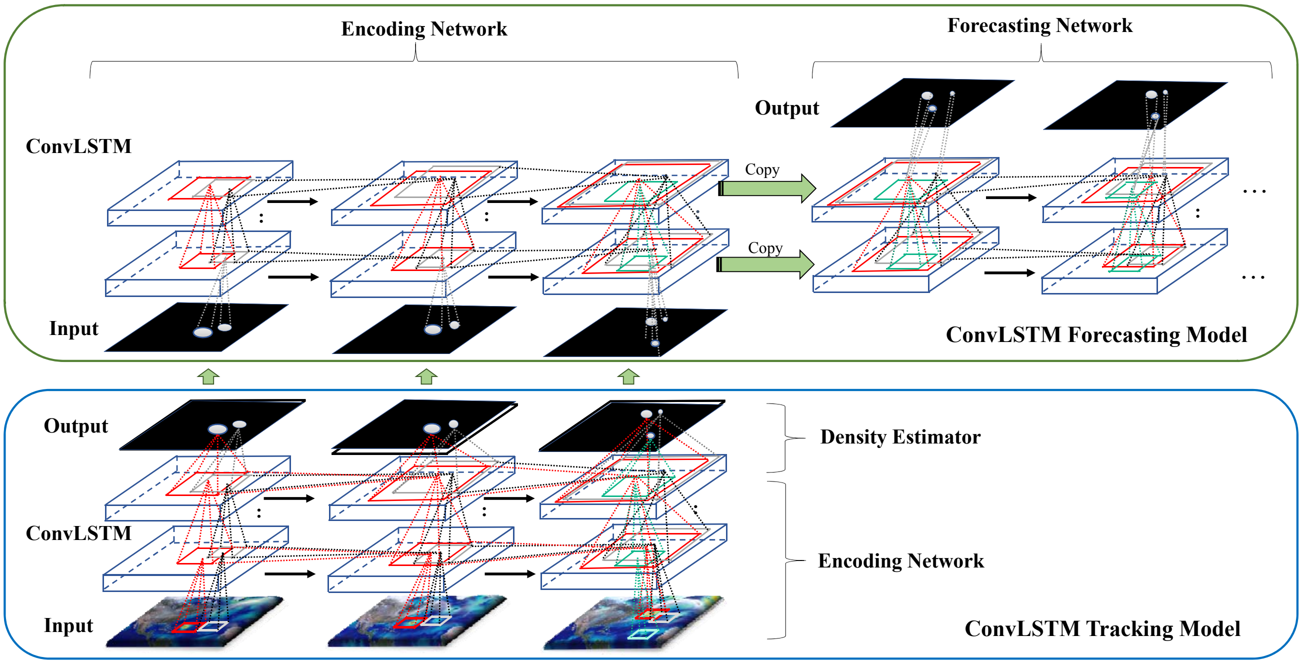


Figure 3. The proposed ConvLSTM-based tracking (*Bottom*) and forecasting (*Top*) model.

nected to its temporal neighbors. In addition, ConvLSTM takes advantage of spatial locality by applying convolution operation in input-to-state and state-to-state transitions. As illustrated in Figure 2, a hidden state $\mathbf{h}_t$ is updated by convolution from the input $\mathbf{X}_t$ and the previous hidden state $\mathbf{h}_{t-1}$. Each convolutional transition is defined by 2-D kernel which determines the range of input region (input-to-state) and previous state region (state-to-sate) to be considered for the new hidden state in the grid. The input-to-state kernel size determines how much the receptive field of the state will grow spatially. Similarly, the state-to-state kernel size decides how much the receptive field of states will grow as time advances (temporally). For example, with an $n \times n$ kernel for state-to-state transition, one pixel in the grid of state will be updated by considering neighbor pixels in $n \times n$ region of previous state by convolution operation. Therefore, the state-to-state transition kernel should be large enough to cover the movement of target object per one time step, and the input-to-sate transition kernel size should be large enough to contain the entire target object.

Figure 3 (*Bottom*) shows our ConvLSTM model for hurricane tracking. Our model consists of two parts. First, the *encoding network* $g : \mathbb{R}^{m\times n\times c\times T} \to \mathbb{R}^{m\times n}$ encodes the input sequence $\mathbf{X}_{1:T}$ to a hidden representation $\mathbf{h}_{1:T}$. The encoding network g may contain L levels of convolution layers, denoted as $\mathbf{h}^{(l)}$ with $l = 1, \cdots, L$ (each layer in the encoding network part in Figure 3). With more levels, each output pixel can be affected by wider range from the input image. For instance, with $k\times k$ kernels with 2 layers, an output pixel is determined by $k \times k$ pixels in the upper hidden layer, which are determined by surrounding $k \times k$ pixels in input image, growing the receptive field in $O(k^2)$. The second part is *density estimator* $f : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$, taking as input $\mathbf{h}_i^{(L)}$, the last hidden layer at time t, and producing the estimated density map $\hat{\mathbf{Y}}_t$. Connecting them together, the output density map at step t is given by

$$\begin{aligned}\hat{\mathbf{Y}}_t &= f(\mathbf{h}_t^{(L)}) \\ &= f(g(\mathbf{X}_1, \cdots, \mathbf{X}_t)) \\ &\approx \arg\max_{\mathbf{Y}_t} p(\mathbf{Y}_t | \mathbf{X}_1, \cdots, \mathbf{X}_t).\end{aligned} \tag{1}$$

We use 5×5 input-to-state and state-to-state kernels, considering the size of typical hurricane (around 250 km or 2.25° of diameter, or 4.5 pixels on our data resolution) and their typical traveling speed (up to 280 km/h, $2.5°/h$, or 5 pixels per 3 hours, which is the unit of measurement in our dataset.

The input climate video consists of 3 channels while the output density-map consists of a single channel. Therefore, to relate the feature maps to density map, we adopt filters of size 3×1 at the density estimator f. We minimize the pixel-wise mean squared loss between the estimated density-map $\hat{\mathbf{Y}}_{1:T}$ and the ground-truth density-map $\mathbf{Y}_{1:T}$:

$$\min_{\Theta} \frac{1}{T} \sum_{t=1}^{T} \frac{1}{mn} \|\hat{\mathbf{Y}}_t - \mathbf{Y}_t\|_2^2, \tag{2}$$

where Θ is a set of parameters in our models f and g.

4. Future Prediction of Hurricane Trajectories

In this section, we take a step further to apply the model from tracking to predicting future trajectory of the target.

4.1. Problem Formulation

We model hurricane forecasting problem as a sequence-to-sequence (seq2seq) forecasting problem in which both the input and the output prediction are spatio-temporal density-map sequences. The input is ground-truth hurricane density-maps of the past T time steps, and the output is predicted hurricane density-maps of the next τ time steps. Formally, given an input sequence $\mathbf{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_T\}$, the model aims to predict a sequence of the most plausible future density-maps $\{\mathbf{Y}_{T+1}, \mathbf{Y}_{T+2}, \cdots, \mathbf{Y}_{T+\tau}\}$, where each $\mathbf{Y}_i \in \mathbb{R}^{m\times n}$ is a fixed size hurricane density-map. Combined with the tracking model in Section 3, we first estimate $\mathbf{Y}_i$ from $\mathbf{X}_i$ for $i = 1, \cdots, T$, and input them to the prediction model described below. This is illustrated in Figure 3.

4.2. Proposed Model

We use encoding and forecasting structures deploying seq2seq architecture using ConvLSTM, similar to the architecture in [17]. Similarly to the tracking model in Section 3, the *encoding network* $\tilde{g} : \mathbb{R}^{m\times n} \rightarrow \mathbb{R}^{m\times n}$ takes a sequence of density maps $\mathbf{Y}_1, \cdots, \mathbf{Y}_T$ as input, and encode them into hidden states $\mathbf{h}_1, \cdots, \mathbf{h}_T$, compressing the entire precedent input density maps into a hidden state. The only difference from tracking is that we take density maps $\mathbf{Y} \in \mathbb{R}^{m\times n}$ instead of the raw climate variables $\mathbf{X} \in \mathbb{R}^{m\times n\times c}$ as input. This is illustrated in Figure 3 (*Top*, the left part).

Once the encoding is done, the hidden layers $\mathbf{h}_T^{(1)}, \cdots, \mathbf{h}_T^{(L)}$ at the last time step T are copied to the *forecasting network* $\tilde{f}$, performing two tasks: 1) state-to-state transition on the hidden states by τ times, the number of desired steps to forecast, and 2) density estimation from the last hidden layers $\mathbf{h}_{T+1}^{(L)}, \cdots, \mathbf{h}_{T+\tau}^{(L)}$to $\hat{\mathbf{Y}}_{T+1}, \cdots, \hat{\mathbf{Y}}_{T+\tau}$. Putting them together,

$$\begin{aligned}
&\widehat{\mathbf{Y}}_{T+1}, ..., \widehat{\mathbf{Y}}_{T+\tau}\\
&= \underset{\mathbf{Y}_{T+1:T+\tau}}{\arg\max}\, p(\mathbf{Y}_{T+1}, ..., \mathbf{Y}_{T+\tau} | \mathbf{Y}_1, ..., \mathbf{Y}_T)\\
&\approx \underset{\mathbf{Y}_{T+1:T+\tau}}{\arg\max}\, p(\mathbf{Y}_{T+1}, ..., \mathbf{Y}_{T+\tau} | g(\mathbf{Y}_1, ..., \mathbf{Y}_T))\\
&= \tilde{f}(\tilde{g}(\mathbf{Y}_1, \cdots, \mathbf{Y}_T)). \qquad (3)
\end{aligned}$$

Considering typical hurricane size and travel speed, we use 5×5 input-to-state and state-to-state kernels, similar to the tracking model. We minimize the pixel-wise mean squared error between predicted and ground-truth density maps:

$$\min_{\Theta} \frac{1}{\tau} \sum_{t=1}^{\tau} \frac{1}{mn} \|\hat{\mathbf{Y}}_{T+1:T+t} - \mathbf{Y}_{T+1:T+t}\|_2^2, \qquad (4)$$

where Θ is a set of parameters of our models $\tilde{f}$ and $\tilde{g}$.

5. Experiment

We evaluate our proposed models on large-scale climate data, both for tracking and forecasting tropical cyclones (hurricanes) from global maps of the climate video. We start by describing the experimental design, followed by results and discussion for both tasks.

5.1. Dataset

We use 20-year records from 1996 to 2015 of the *Community Atmospheric Model v5 (CAM5)* dataset. It contains snapshots of the global atmospheric states every 3 hours. Each snapshot is comprised of multiple physical variables, among which we use zonal wind (U850), meridional wind (V850), and precipitation (PRECT), given their relevance to hurricane identification from scientific studies. We use the corresponding TECA labels [16] for ground truth, represented as density maps of hurricane trajectories as described in Section 3.1. The TECA labels contain spatial coordinate (latitude, longitude) of each hurricane and the diameter of hurricane-force winds. We synthesize the ground-truth density maps based on Gaussian mixtures.

From this dataset, we created 11,160 sub-sequences for training examples of length 10, corresponding to 30 hours. For tracking, we use 80% for training, 10% for validation, and the other 10% for testing. For forecasting, we use the first 5 frames as input and test on the next 5 frames.

In order to fit the model into memory, we split the global map into several non-overlapping tropical cyclone basins[1] of $60° \times 160°$ sub-images ($1° \approx 111\ km$). Also, we reduce the original resolution of CAM5 by 2×2 max pooling. After reduction, the input image size is 128×256 with around $0.5°$ ($55.5\ km$) resolution.

5.2. Baselines

To train the baseline detection CNN (Section 3.2), we collected 96,733 climate image patches sized 20×20 (representing $5° \times 5°$, corresponding to $555\ km$) centered on hurricanes for positive examples. We collected the same amount of random places with same size for negative examples. We also used the TECA labels for this model. The pretrained detection model achieved test accuracy of 99.1%.

The detection CNN model consists of 6 layers, including 4 convolutional and 2 fully connected layers. Each convolutional layer has 64 5×5 filters, followed by a ReLU activation and a 2×2 max pooling. We combined all variables

[1] Typically, there are seven commonly accepted basins, including North Atlantic, Northeast Pacific, Northwest Pacific, North Indian, South Indian, South Pacific, and South Atlantic. Due to the local environment, tropical cyclones do not cross the border of these basins, that is, hurricanes always occur, develop, and disappear within the same basin.

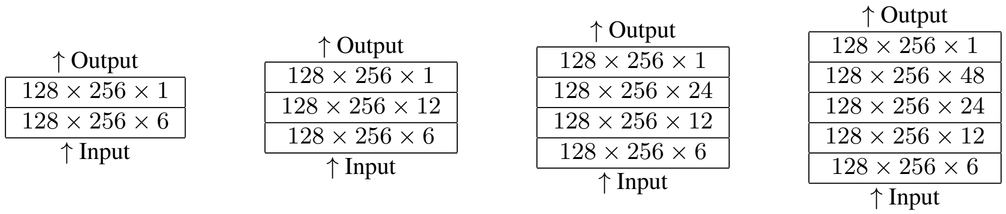


Figure 4. ConvLSTM architectures used in tracking experiments. From left to right, $\{2, 3, 4, 5\}$-layer model.

before feeding them into the model.[2] After convolutional layers, softmax activation is used for the final output.

For 3D Encoder-Decoder model [15], bounding-boxes of extreme climate events are constructed from the spatio-temporal climate simulation data with 8 time steps (24 hours) and 16 variables.

5.3. Tracking Experiment

We conduct both quantitative and qualitative studies, comparing our ConvLSTM-based tracking model against several baselines.

5.3.1 Experimental Settings

For ConvLSTM tracking model, we test with four architectures, illustrated in Table 4. All the input-to-state and state-to-state kernels are of size 5×5. The batch size is 24, and all ConvLSTM layers have forget bias of 1. We use AdaGrad optimizer with learning rate of 0.001. We apply 10% dropout during training. We initialize all the states of the ConvLSTM to zero before the first input comes, meaning that no information has been memorized in cell from the past. For training and testing, we utilized 4 GPUs (NVIDIA Tesla K80).

Our model outputs a density map, while the ground truth is given by a point (hurricane center) or by a bounding box (within the diameter of hurricane-force winds from the center). To fill in this gap, we first take pixels above a threshold θ and cluster them. Among the pixels belonging to the same cluster, we take the maximum peak as its center. To be consistent with Racah et al. [15], we take a bounding box of size 32×32 (1,776 km × 1,776 km) centered on the maximum peak. If the *Intersection of Union (IOU)*, the ratio of overlapping areas to the union of two bounding boxes, between the bounding box of our prediction and that of the ground truth is higher than pre-defined thresholds (0.1 and 0.5 according to [15]), the prediction is marked as matched. By counting the number of matched clusters (TP) versus the total number of detected clusters in ground truth ($TP + FN$) and output ($TP + FP$), precision ($\frac{TP}{TP+FP}$) and recall ($\frac{TP}{TP+FN}$) are obtained, respectively. With multiple thresholds $0 < \theta < 1$, we generate the precision-recall curve, from which we compute the average precision.

Model	IOU: 0.1	IOU: 0.5
Detection-based CNN [7]	14.78%	12.26%
3D Conv Encoder-decoder [15]	22.65%	15.53%
2 layered ConvLSTM	88.96%	87.32%
3 layered ConvLSTM	$\mathbf{93.77}\%$	$\mathbf{92.18}\%$
4 layered ConvLSTM	92.58%	90.87%
5 layered ConvLSTM	93.49%	91.93%

Table 1. Comparison of AvgP for different tracking models.

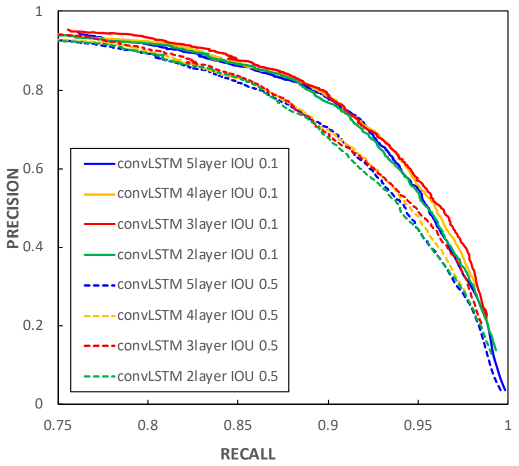


Figure 5. Precision and Recall of ConvLSTM models with IOU threshold 0.1 (solid lines) and 0.5 (dotted lines). We draw with same color for the results from same architecture.

5.3.2 Result and Discussion

The density maps generated by the proposed models contain a wide range of values, so we vary the threshold θ from 0.001 to 0.97 with interval of 0.001 to trade-off precision and recall. Figure 5 presents the precision-recall curves of the ConvLSTM models with 2 to 5 layers at IOU threshold 0.1 and 0.5. We observe that deeper models perform consistently better than the 2-layer model for our dataset, but it is not clear to benefit from more than 3 layers. On the other hand, we clearly observe that the lower IOU threshold (0.1) performs better than the other (0.5), probably because more

[2] We also tried separate layers for each variable combined later (late fusion), but early fusion worked better.

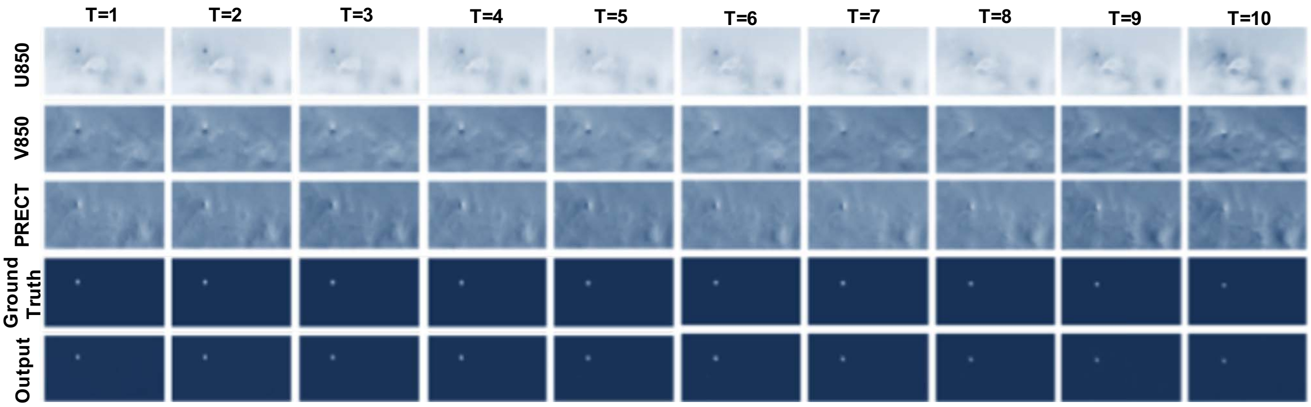


Figure 6. Results from ConvLSTM tracking model. From the top, input channels (U850, V850, PRECT), ground truth, and output (frame size: 60° × 160°). See Figure 7 for the magnified view around the hurricane.

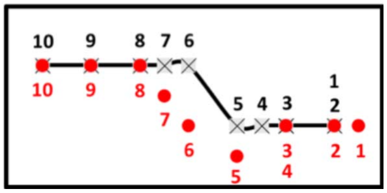


Figure 7. Magnified view of tracking result in Figure 6 to frame size 3° × 8°. Black line is ground truth, and red dots are predicted trajectories from our ConvLSTM-based tracking model.

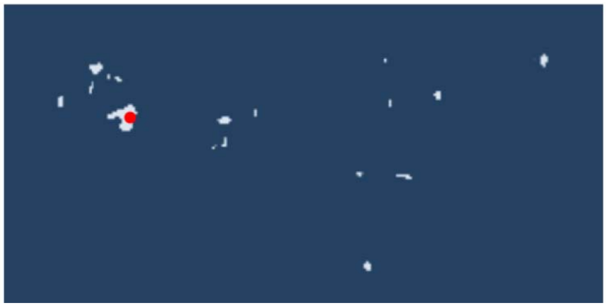


Figure 8. An output sample from detection-based CNN tracking model, showing many false positives. Other than the ground truth hurricane location (red dot), all other detected white pixels are false positives. (Frame size is same as each image in Figure 6, 60° × 160°.)

events are marked as true positive at IOU threshold 0.1 than at 0.5.

Table 1 compares our ConvLSTM-based tracking model against two baselines; a variant of detection-based CNN model [7] and 3D Encoder-Decoder model [15], in terms of **Average Precision (AvgP)**, defined as:

$$\text{AvgP} = \int_0^1 p(r)dr \approx \sum_k p(k)\Delta r(k), \qquad (5)$$

where $p(r)$ is precision at recall level r, equivalent to the area under the curve from Figure 5. The detection-based CNN model [7] achieves a high recall, but a significantly lower precision due to many false positives. Therefore, the average precision is below 15% for both test cases at IOU threshold 0.1 and 0.5. The 3D Encoder-Decoder model [15] performs slightly better, but still below 23%. In contrast, our model achieves a far higher average precision above 90% with both IOU thresholds tested. From this experiment, we verify that the proposed ConvLSTM-based method significantly outperforms the two baseline methods.

Figure 6 and 7 shows example results from our ConvLSTM-based tracking model. Qualitatively, we observe that the ConvLSTM-based model produces density-maps that are closest to the ground truth (Figure 6) and obtained trajectory based on density maps from the proposed model is in strong agreement with ground truth trajectory (Figure 7). In contrast, the CNN-based model provides a rough estimation with many false positives, as shown in Figure 8.

Our quantitative and qualitative results indicate that the proposed tracking model using ConvLSTM is suitable for capturing spatio-temporal patterns of hurricane, mainly because the proposed ConvLSTM can capture a detailed latent structure of the extreme climate event by learning not only pixel-level spatial dynamics between multiple climate variables but also long-term temporal dynamics.

We conjecture that the errors in the ConvLSTM tracking model mainly come from 1) *miss-match between density-map and hurricane segmentation*; the starting and finishing period of the detected hurricane may not be circular as opposed to our assumption, and the Gaussian heat-map is not sufficient to catch them, and 2) *lack of information*; only with the three variables we used, it might not be possible to correct most false positives as they show similar patterns of high wind and precipitation with the true hurricanes. We may need more climate variables to reduce these false positives.

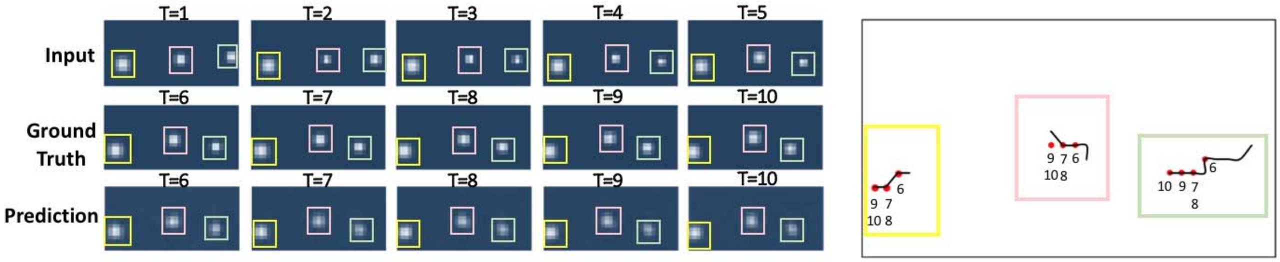


Figure 9. *Left*: An example result of prediction model (size of frame: 30° × 20°). *Right*: Comparison between the ground truth and predicted trajectory. Black: ground truth (time 1 to 10), Red: prediction (time 6 to 10); size of frame 30° × 20°.

RMSE	$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$
(in pixels)	0.48	2.54	2.68	2.68	3.07
(in km)	26.64	140.97	148.74	148.74	170.38

Table 2. Prediction accuracy for each future time steps. We see that our model accurately predicts the hurricane location after 3 hours ($\tau = 1$), while error gets larger for longer future as expected.

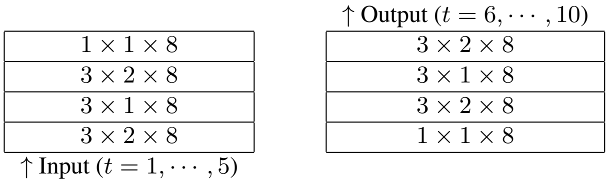


Figure 10. ConvLSTM architecture used in forecasting experiments. *Left*: encoding network, *Right*: forecasting network.

5.4. Forecasting Experiment

5.4.1 Experimental Settings

Our forecasting model consists of the encoding network and the forecasting network both with 4-layered ConvLSTM, as illustrated in Table 10. Considering simple Gaussian pattern of density maps, we design the size of all input-to-state and state-to-state kernels as 3 × 3. Batch size is 16, and all ConvLSTM layers have forget bias of 1. Similar to the tracking model, all states are initialized to zeros, and optimized with AdaGrad with learning rate of 0.001. We apply 10% dropout during training.

We minimize the pixel-wise mean squared loss between the output and the corresponding frames (last 5 frames among 10) in ground truth. We evaluate results qualitatively by visual comparison of the prediction with the ground truth, and quantitatively by computing root mean squared error (RMSE) between the ground truth hurricane center and the predicted hurricane center at each future time step.

5.4.2 Result and Discussion

The experiment demonstrates that the proposed model precisely predicts location of hurricanes after 3 hours ($\tau = 1$), with less than 30 km of errors, as shown in Table 2. After 6 to 15 hours ($\tau = 2$ to 5), the average error is approximately 140 to 170 km.

Figure 9 shows an example forecasting result with our model. The left plot shows an input sequence of 5 time steps (top row), ground truth (middle row), and predicted (bottom row) next 5 time steps. The right plot combines predicted sequence (red dots) with ground truth path (black line), connecting hurricane centers at each time frame. We can clearly see that our model correctly predicts three hurricanes moving from right to left. Despite low resolution of the density map, the predicted trajectories match reasonably well with respect to the ground truth.

6. Conclusion

In this paper, we address unique challenges in representation learning of extreme climate events using neural networks, and propose best performing models to track and predict hurricane trajectories. Specifically, we propose the regression-based tracking model using ConvLSTM and show our proposed model significantly outperforms in detecting hurricane from spatio-temporal climate simulation data by comparing detection accuracy with existing baselines. In addition, we present the forecasting network deploying seq2seq model based on our ConvLSTM model.

Future directions of work to improve accuracy of the proposed ConvLSTM model may include 1) training different models for different hurricane basins to capture distinctive trajectory patterns, 2) utilizing time-sequential image segmentation for ground truth instead of synthesized density maps, and 3) building a larger-scale dataset including more physical variables and longer time sequences without sacrificing resolution.

Acknowledgement

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. LLNL-PROC-763617. This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

References

[1] S. Alemany, J. Beltran, A. Perez, and S. Ganzfried. Predicting hurricane trajectories using a recurrent neural network. In *arXiv:1802.02548v2*, 2018. 2

[2] L. Dong and F. Zhang. Obest: An observation-based ensemble subsetting technique for tropical cyclone track prediction. *Weather and Forecasting*, 31(1):57–70, 2016. 2

[3] R. L. Elsberry, J. R. Hughes, and M. A. Boothe. Weighted position and motion vector consensus of tropical cyclone track prediction in the western north pacific. *Monthly Weather Review*, 136(7):2478–2487, 2008. 2

[4] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[5] W. Kim and O. Hasegawa. Time series prediction of tropical storm trajectory using self-organizing incremental neural networks and error evaluation. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 22(4):465–474, 2018. 2

[6] T. Kurth et al. Deep learning at 15PF: supervised and semi-supervised classification for scientific data. In *Proc. of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2017. 1

[7] Y. Liu et al. Application of deep convolutional neural networks for detecting extreme weather in climate datasets. *ArXiv:1605.01156*, 2016. 1, 2, 3, 6, 7

[8] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[9] A. C. Lorenc. Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112(474):1177–1194, 1986. 1

[10] W. Luo, X. Zhao, and T. Kim. Multiple object tracking: A review. *ArXiv:1409.7618*, 2014. 3

[11] S. J. Majumdar and P. M. Finocchio. On the ability of global ensemble prediction systems to predict tropical cyclone track probabilities. *Weather and Forecasting*, 25(2):659–680, 2010. 2

[12] S. V. Poroseva, N. Lay, and M. Y. Hussaini. Multimodel approach based on evidence theory for forecasting tropical cyclone tracks. *Monthly Weather Review*, 138(2):405–420, 2010. 2

[13] Prabhat, S. Byna, V. Vishwanath, E. Dart, M. Wehner, W. D. Collins, et al. TECA: Petascale pattern recognition for climate science. In *Proc. of the International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2015. 2

[14] L. Qi, H. Yu, and P. Chen. Selective ensemble-mean technique for tropical cyclone track forecast by using ensemble prediction systems. *Quarterly Journal of the Royal Meteorological Society*, 140(680):805–813, 2014. 2

[15] E. Racah, C. Beckham, T. Maharaj, S. E. Kahou, Prabhat, and C. Pal. ExtremeWeather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2, 6, 7

[16] O. Rübel, S. Byna, K. Wu, F. Li, M. Wehner, W. Bethel, et al. TECA: A parallel toolkit for extreme climate analysis. *Procedia Computer Science*, 9:866–876, 2012. 2, 5

[17] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 2, 3, 5

[18] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2

[19] J. A. Sippel and F. Zhang. A probabilistic analysis of the dynamics and predictability of tropical cyclogenesis. *Journal of the Atmospheric Sciences*, 65(11):3440–3459, 2008. 2

[20] A. D. Snyder, Z. Pu, and Y. Zhu. Tracking and verification of east atlantic tropical cyclone genesis in the ncep global ensemble: Case studies during the NASA African Monsoon multidisciplinary analyses. *Weather and Forecasting*, 25(5):1397–1411, 2010. 2

[21] C. Thanh, T. T. Tien, and K. Q. Chanh. Application of breeding ensemble to tropical cyclone track forecasts using the regional atmospheric modeling system (rams) model. *Applied Mathematical Modelling*, 40(19–20):8309–8325, 2016. 2

[22] T. T. Tien, C. Thanh, H. T. Van, and K. Q. Chanh. Two-dimensional retrieval of typhoon tracks from an ensemble of multimodel outputs. *Weather and Forecasting*, 27(2):451–461, 2012. 2

[23] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. *ArXiv:1706.08033*, 2017. 3

[24] H. C. Weber. Hurricane track prediction using a statistical ensemble of numerical models. *Monthly Weather Review*, 131(5):749, 2003. 2

[25] F. Xiong, X. Shi, and D.-Y. Yeung. Spatiotemporal modeling for crowd counting in videos. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 3