Learning to Focus and Track Extreme Climate Events

Sookyung Kim[*1]
kim79@llnl.gov
Sunghyun Park[*2]
psh01087@korea.ac.kr
Sunghyo Chung[*3]
shawn.chung@kakaocorp.com
Joonseok Lee[4]
joonseok@google.com
Yunsung Lee[2]
swack9751@korea.ac.kr
Hyojin Kim[1]
hkim@llnl.gov
Mr Prabhat[5]
prabhat@lbl.gov
Jaegul Choo[2]
jchoo@korea.ac.kr

[1] Lawrence Livermore Nat'l Lab.
Livermore, CA, USA
[2] Korea University
Seoul, Korea
[3] Kakao Corp.
Seongnam, Korea
[4] Google Research
Mountain View, CA, USA
[5] Lawrence Berkeley Nat'l Lab.
Berkeley, CA, USA

Abstract

This paper tackles the task of extreme climate event tracking. It has unique challenges compared to other visual object tracking problems, including a wider range of spatio-temporal dynamics, the unclear boundary of the target, and the shortage of a labeled dataset. We propose a simple but robust end-to-end model based on multi-layered ConvLSTMs, suitable for climate event tracking. It first learns to imprint the location and the appearance of the target at the first frame in an auto-encoding fashion. Next, the learned feature is fed to the tracking module to track the target in subsequent time frames. To tackle the data shortage problem, we propose data augmentation based on conditional generative adversarial networks. Extensive experiments show that the proposed framework significantly improves tracking performance of a hurricane tracking task over several state-of-the-art methods.

1 Introduction

Tracking climate events are pressing and challenging problems that humanity has faced for a long time. Traditionally, most conventional approaches have been built upon human expertise based on scientific intuition and related physics variables, such as wind speed and humidity. Human experts tracked climate events by associating similar events in two consecutive frames based on manually chosen patterns [22] in a high-resolution climate image

[*] These authors contributed equally.

(a multi-channel 2-D matrix with climate variables, such as wind speed and surface pressure, rather than an RGB color density) sequences from physics-based simulations.

Recently, computer vision community has made significant progress by applying various pattern recognition techniques in visual object tracking, a task to locate a target object in a video, maintaining its identity and yielding its individual trajectory, given its initial location in the first frame. **Extreme climate event tracking** is similar to visual object tracking, but it has unique, challenging aspects:

1. Climate events may be dependent on *longer-term* and *wider-range* spatio-temporal dynamics (known as a 'butterfly effect') between multiple scientific variables than the targets in visual object tracking do on RGB pixels.
2. The target events are not often defined as rigid bodies, *flexibly changing their shape with no clear boundary*, and are *difficult to visually distinguish* them from each other. Thus, it is generally difficult to associate an object of interest with the correct one in consecutive frames.
3. Climate event data are usually *sparsely collected* (both *temporally* and *spatially*), as it occurs rarely but requires special devices installed in the wild. For instance, the hurricane data we use in this paper are collected for every three hours, a sufficient time for a hurricane to move 350 km. This makes it difficult for us to assume the object would appear nearby in consecutive frames.

Because of these challenges, a conventional tracking-by-detection method, which relies heavily on the amount of training data and detects the object mainly by its appearance but relatively neglects spatio-temporal dynamics, is less suitable for this problem. An ideal climate event tracker needs to effectively take long-term and wide-range dynamics into account, capturing subtle differences among events from sparsely collected training data.

In this work, we propose a simple but robust end-to-end model, suitable for the climate event tracking problem. Specifically, the proposed model consists of two sub-modules, the **(1) focus learning module** to learn where and what to focus, and the **(2) tracking module** to track what we focused on. The focus learning module is designed to extract the latent feature of the target event from the first frame, given the initial image and the location of the target. Given the representation of the target event, the tracking module predicts its location by localizing the learned feature of a target object in the subsequent frames.

In both modules, we adopt ConvLSTM-based auto-encoder structure for the two following reasons. First, it learns a mapping from time-series climate variables to a time-series density map with event probabilities, suffering less from a blurry boundary of climate events. It compactly embeds the target's historical appearance change and movement in hidden states, capturing essential spatio-temporal dynamics of the target. This addresses the property 2 above. Second, we can easily adjust the receptive field of the ConvLSTM with a larger kernel, taking broader spatio-temporal information into account to capture dynamics of the target event, which addresses the property 1.

In addition, to tackle the problem of the limited training data for hurricane tracking (property 3), we adopt a state-of-the-art data augmentation technique based on conditional generative adversarial networks (GANs) to synthesize plausible labeled hurricane videos given our existing labeled data. Our experiment indicates that training our model with synthetic hurricane data in addition to real data significantly improves tracking accuracy.

2 Related Work

Given the location of the target object in the initial frame, visual object tracking localizes the target in subsequent frames. Recently, various neural-net-based tracking models were proposed.

Tracking-by-detection approaches first detect candidates of the target mainly by using convolutional neural networks (CNNs). Afterwards, the most probable candidate is chosen (an association step). When multiple target objects exist, each target needs to be associated with a distinct candidate. Among various CNN-based trackers, multi-domain CNN (MDNet) [13, 24] achieves the state-of-the-art performance on multiple datasets by learning discriminative features for instance classification. Other tracking approaches, such as Siam-FC [2], SA-Siam [10], and Siamese-RPN [20], utilize Siamese networks to associate the objects based on the highest similarity score obtained by exhaustively testing all the possible locations in consecutive frames. There have been efforts to combine the online learning efficiency of the correlation filter (CF) [3] with the discriminative power of CNN features trained offline. In particular, ECO [5] and C-COT [4] achieve the state-of-the-art performance in its accuracy in multiple standard tracking benchmark datasets such as VOT [19] and OTB [40].

To track a target without a rigid boundary or motion against the background, pixel-wise object tracking methods have advantages over tracking-by-detection approaches. Son et al. [34] adopted gradient-boosted decision trees to estimate pixel-level annotation of a segmentation mask. Jang et al. [12] proposed the encoder-decoder model to use deconvolution for image segmentation and contour detection. Yeo et al. [42] proposed to utilize a Markov chain approach on a super-pixel graph. Recently, convolutional LSTM (ConvLSTM) [30] has been widely applied to handle spatio-temporal dynamics in analyzing video data, e.g., precipitation forecasting [31], motion prediction [8], pixel-wise video prediction [38], crowd counting [41], and segmentation-based tracking [35]. Romera et al. [28] used ConvLSTMs for instance segmentation on a single image.

Detecting and predicting extreme climate events. Conventional extreme climate event detection and tracking methods rely on a number of numerical simulation-based methods, including an ensemble of multiple prediction models or multi-scale prediction systems [6, 7, 23, 25, 26, 32, 33, 36, 37, 39]. Recently, with large-scale datasets, climate research communities have started to leverage various deep learning techniques. The extreme climate event detection and localization problem was tackled with recurrent neural networks (RNNs) [1] and spatio-temporal CNNs [27]. Also, ConvLSTM [16] and incremental neural networks [18] were proposed to predict the future trajectory of hurricanes and cyclones. Region CNNs were applied to classify different types of extreme climate events [14, 15, 21]. Kim et al. [17] predicted the concentration of air pollutants using LSTMs.

Most existing approaches, however, have not addressed unique challenges to handle sparse climate data covering a wide geographic range for an extended period. In this paper, we tackle the unique challenges of the climate event tracking problem introduced in Section 1 with the ConvLSTMs-variant model, which is specially designed to capture a wide range of spatio-temporal dynamics, as well as with a novel data augmentation based on generative adversarial networks (GANs).

3 Problem Formulation

Notations. We denote $\mathbf{X} = \{\mathbf{X}_0, \mathbf{X}_1, ..., \mathbf{X}_{T-1}\}$ as a climate video of length T, where $\mathbf{X}_i \in \mathbb{R}^{m \times n \times c}$ for $i = 0, \ldots, T-1$ is a 2-D climate image of size $m \times n$ with c climate channels (e.g., surface-level pressure and wind speed). $\mathbf{X}$ may contain multiple trajectories of target events (e.g., hurricanes), starting and ending at different frames, but our approach tracks one target event at a time. The ground truth $\mathbf{y} = \{\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_{T-1}\}$ is the location of the target climate event, where $\mathbf{y}_i \equiv \{x_i, y_i, w_i, h_i\}$ for $i = 0, ..., T-1$ represents the bounding box of the target event, i.e., (x_i, y_i) being the top-left position, w_i and h_i being the width and the height, respectively.

Extreme Climate Event Tracking. Given a climate video $\mathbf{X}$ and the initial location of the target event $\mathbf{y}_0$, the goal is to estimate its locations $\hat{\mathbf{y}}_i$ in subsequent image frames as closely as possible to the ground truth $\mathbf{y}_i$ for $i = 1, ..., T-1$.

4 Proposed Method: Focus-and-Track Framework

Given a climate video $\mathbf{X}$ and an initial location of the target object $\mathbf{y}_0$, our framework aims at predicting the trajectory of the target object. It is tempting to directly regress bounding box elements from the input image $\mathbf{X}_i$, but several issues exist. First, as the boundary between the target event and the background is often blurry, the accurate regression of the bounding box $(\hat{\mathbf{y}}_i)$ from $\mathbf{X}_i$ is challenging. Second, when multiple events exist in the frame, data association is difficult as their appearances are often too similar to distinguish visually.

To address these challenges, we represent both ground truth and prediction as density maps. That is, each ground truth label $\mathbf{y}_i$ is transformed to a density map $\mathbf{H}_i \in \mathbb{R}^{m \times n}$ with Gaussian mixtures $\mathcal{N}(\mathbf{y}_i, \sigma^2 \mathbf{I})$, where the variance σ^2 is determined by the hurricane radius.[1] Then, we model the tracking problem as pixel-wise regression at each time step, minimizing the pixel-wise mean squared error between the ground truth density map $\mathbf{H}_i$ and our prediction output $\hat{\mathbf{H}}_i \in \mathbb{R}^{m \times n}$. That is, we predict the probability for the target to be observed in each pixel as $\hat{\mathbf{H}}_i$. Once obtaining $\hat{\mathbf{H}}$, we regress it to the original bounding box $\hat{\mathbf{y}}$.

Model Overview. AS shown in Figure 1, Our framework consists of two modules: (1) the *focus learning module* that learns to extract the latent features of the target object at the initial time step, and (2) the *tracking module* to estimate the bounding box information of the target trajectory in subsequent time frames. Specifically, the focus learning module takes the initial frame of the climate video ($\mathbf{X}_0$) focusing only where the target is (that is, all other places are masked out using $\mathbf{H}_0$), and estimates the density map of the target $\hat{\mathbf{H}}_0$. That is, the focus learning module learns a mapping from the focused target to the density map, encoding the appearance and the location of the target object in its hidden state. Next, the tracking module takes the learned feature in the hidden state of the focus learning module (weight sharing of convLSTM cell) and learns to locate the target in subsequent frames. We use a many-to-many RNN architecture using ConvLSTM cells.

4.1 Focus Learning Module

This module takes the initial frame with its target location, learns to represent the target, and provides this compact representation to the tracking module. Figure 1 (*left*) shows our focus

[1] We chose a diagonal covariance matrix because most hurricanes are in a circular shape. For other types of extreme climate events, e.g., atmospheric river, a full covariance matrix may be used.

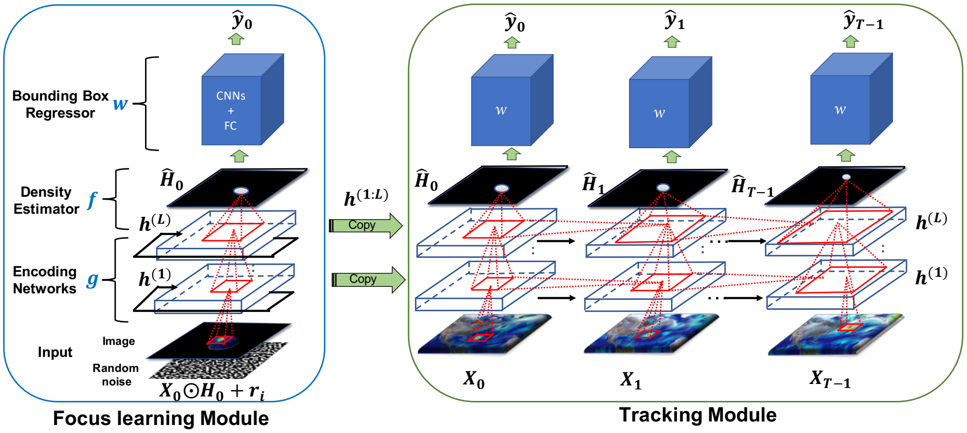

Figure 1: Overview of our proposed focus (*left*) and tracking(*right*) approach.

learning module for event tracking.

Input Pre-processing. Given the initial frame $\mathbf{X}_0 \in \mathbb{R}^{m \times n \times c}$ of the climate video and the same-sized density map $\mathbf{H}_0 \in \mathbb{R}^{m \times n}$ of the target, we first take their element-wise multiplication ($\mathbf{X}_0 \odot \mathbf{H}_0$), masking out the frame except for where the target is.

To make the model stably track the target in the latter (tracking) stage even with some variation in its appearance and location, we create N slightly different images of $\mathbf{X}_0 \odot \mathbf{H}_0$ by adding Gaussian noise $\mathbf{r}_i \in \mathbb{R}^{m \times n}$ for $i = 1, 2, ..., N$ and translation perturbation. In this manner, the model learns the latent representation of the target that is robust to noise and invariant to translation.

Encoding Networks. The N variations of the masked initial frame $\mathbf{X_0} \odot \mathbf{H_0}$ are taken as a sequence input to the L-layered ConvLSTM *encoding networks* $g : \mathbb{R}^{m \times n \times c} \to \mathbb{R}^{m \times n \times k_L}$, which embeds each image to a fixed-dimensional hidden representations $\mathbf{h}^{(i)} \in \mathbb{R}^{m \times n}$, for $i = 1, ..., L$, where k_i is the number of features in each layer. Note that it is important to use a sufficiently large number of layers and receptive fields to capture a long and wide range of spatio-temporal dynamics of climate variables.

Density estimator. The hidden state $\mathbf{h}^{(L)}$ from the last layer of the encoding networks is fed into the *density estimator* $f : \mathbb{R}^{m \times n \times k_L} \to \mathbb{R}^{m \times n}$, estimating a single-channel density map in the first frame $\hat{\mathbf{H}}_0$, such that $\hat{\mathbf{H}}_0 \approx \mathbf{H}_0$.

Integrating the two sub-modules, the output density map is generated as

$$\hat{\mathbf{H}}_0 = f(\mathbf{h}^{(L)}) = f(g(\mathbf{X}_0 \odot \mathbf{H}_0 + \mathbf{r}_1, \cdots, \mathbf{X}_0 \odot \mathbf{H}_0 + \mathbf{r}_N)) \approx p(\mathbf{H}_0 | \mathbf{X}_0 \circ \mathbf{H}_0). \tag{1}$$

At each step i (from 1 to N), the model f repeatedly produces a 2-D density-map $\hat{\mathbf{H}}_0$, based on the input image $\mathbf{X}_0$ with random noise $\mathbf{r}_i$ and the previous hidden state $\mathbf{h}_{i-1}^{(1:L)}$, an encoding of the input so far. As we iterate, the hidden state $\mathbf{h}_i^{(1:L)}$ gradually learns and encodes the feature of the target. After N iterations, the hidden state of the last iteration step, $\mathbf{h}_N^{(1:L)}$, is used as the latent representation of the target in the tracking module. It is worth noting that excluding $\mathbf{X}_0$ and $\mathbf{r}_i$ in Eq. (1), f and g functions as an auto-encoder to extract latent representation based on $\mathbf{H}_0$.

Bounding Box Regressor. Lastly, we regress the original bounding-box ground truth $\mathbf{y}_0$ from the produced density map $\hat{\mathbf{H}}_0$, with a *bounding box regressor* $w : \mathbb{R}^{m \times n} \to \mathbb{N}^4$, where its output consists of the four bounding box elements, $\{x_0, y_0, w_0, h_0\}$. Formally, $\hat{\mathbf{y}}_0 = w(\mathbf{y}_0|\hat{\mathbf{H}}_0)$ such that $\hat{\mathbf{y}}_0 \approx \mathbf{y}_0$. We use multi-layered CNNs followed by a fully connected layer for w.

Loss Functions. To train the model f and g, we minimize L_{gf}, the pixel-wise mean squared loss between the estimated density-map $\hat{\mathbf{H}}_0$ and the ground-truth density-map $\mathbf{H}_0$, averaged over all perturbed images. Similarly, for the model w, we minimize the squared loss L_w between the estimated bounding box elements $\hat{\mathbf{y}}_\mathbf{0}$ and ground truth $\mathbf{y}_\mathbf{0}$, i.e.,

$$\mathcal{L}_{gf} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{mn}\|\hat{\mathbf{H}}_0 - \mathbf{H}_0\|_2^2 \qquad \mathcal{L}_w = \frac{1}{4}\|\hat{\mathbf{y}}_0 - \mathbf{y}_0\|_2^2 \tag{2}$$

The overall loss $\mathcal{L}$ is defined as a weighted (by a constant α) sum of $\mathcal{L}_{gf}$ and $\mathcal{L}_w$, which we minimize end-to-end. That is, $\mathcal{L} = \mathcal{L}_{gf} + \alpha\mathcal{L}_w$.

4.2 Tracking Module

Given the final hidden representations $\mathbf{h}^{(1:L)}$ from the focus learning module as its initial state, the tracking module learns to track the target in subsequent frames of the video. In the tracking module, all weights of the ConvLSTM cell from the focus learning module are shared to update and store the spatio-temporal variation of the target in its hidden states. The main architecture of the tracking module is similar to that of the focus learning module, illustrated in Figure 1 (*right*). The tracking module estimates the density map of the target object, based on the input image $\mathbf{X}_i$ at time step i and the hidden state from previous time steps, $\mathbf{h}_{i-1}^{(1:L)}$, deploying the many-to-many multi-layered ConvLSTM architecture as the encoder g and the density map estimator f. Afterwards, the estimated density map $\hat{\mathbf{H}}_i$ at each time step i is fed into the bounding box regressor w to estimate bounding box elements. Formally, it is written as

$$\hat{\mathbf{H}}_i = f(g(\mathbf{H}_i|\mathbf{X}_i, \mathbf{h}_{i-1}^{(1:L)})) \approx p(\mathbf{H}_i|\mathbf{X}_i, \mathbf{h}_{i-1}^{(1:L)}), \qquad \hat{\mathbf{y}}_i = w(\mathbf{y}_i|\hat{\mathbf{H}}_i). \tag{3}$$

Similar to the focus learning module, the overall loss is defined as the weighted sum of L_{gf} and L_w, trained end-to-end.

4.3 Data Augmentation

To tackle the shortage of labeled training data for hurricane tracking, we adopt a state-of-the-art data augmentation technique to synthesize plausible labeled hurricane video given our existing labeled data. As shown in Figure 2, we first generate a hurricane trajectory (Step 1), and then, given the coordinate of hurricane center in the trajectory, we generate a hurricane image corresponding to the channels with climate variables (Step 2). Key features of the architecture are discussed below.

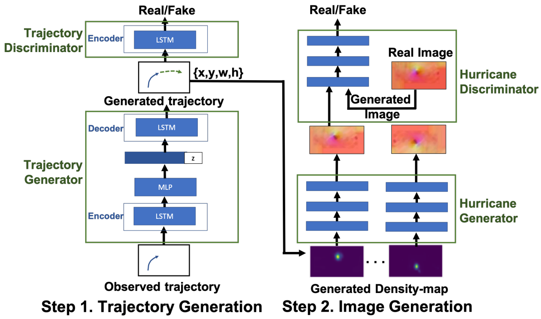

Figure 2: Our data augmentation approach.

Trajectory Generation. The hurricane trajectory generation model is based on Social GAN (SGAN) [9], where the future trajectories of multiple people are predicted simultaneously from trajectories of each person in the scene.

Given an input trajectory of a hurricane $\{\mathbf{y}_1,...,\mathbf{y}_t\}$, our task is generating its future trajectory from $t+1$ to T, where each $\mathbf{y}_i = \{x_i, y_i, w_i, h_i\}$ is a bounding box element. The Generator takes $\{\mathbf{y}_1,...,\mathbf{y}_t\}$ as input and produces $\{\hat{\mathbf{y}}_{t+1},...,\hat{\mathbf{y}}_T\}$. The discriminator takes the entire sequence, either from the real sequence $\{\mathbf{y}_1,...,\mathbf{y}_T\}$ or from the generated one $\{\mathbf{y}_1,...,\mathbf{y}_t,\hat{\mathbf{y}}_{t+1},...,\hat{\mathbf{y}}_T\}$, and classify it as either real or fake.

First, the location and scale of the hurricane is embedded and fed into the encoder LSTMs. That is, the encoder hidden state $\mathbf{h}_e^{(i)}$ for $i = 1,...,t$ is given by

$$\mathbf{h}_e^{(i)} = \mathrm{LSTM}\left(\mathbf{h}_e^{(i-1)}, \phi(\mathbf{y}_i)\right), \tag{4}$$

where ϕ is an embedding function of a bounding box, followed by a multi-Layer perceptron (MLP) γ. Next, we concatenate the output of γ with a Gaussian noise $\mathbf{z} \sim \mathcal{N}(0,\mathbf{I})$ to initialize the hidden state of the decoder $\mathbf{h}_d^{(t)}$:

$$\mathbf{h}_d^{(t)} = \left[\gamma(\mathbf{h}_e^{(t)});\mathbf{z}\right] \tag{5}$$

After initializing the decoder hidden state $\mathbf{h}_d^{(t)}$, we predict the next coordinate of trajectory $\hat{\mathbf{y}}_{i+1}$ from its previous coordinate $\hat{\mathbf{y}}_i = \{x_i, y_i, w_i, h_i\}$ for $i = t,...,T-1$ as

$$\mathbf{h}_d^{(i+1)} = \mathrm{LSTM}\left(\mathbf{h}_d^{(i)}, \phi(\mathbf{y}_i)\right), \quad \hat{\mathbf{y}}_{i+1} = \gamma(\mathbf{h}_d^{(i+1)}) \tag{6}$$

where ϕ is an embedding function and γ is an MLP.

Image Generation. We adapt the image generation from the pix2pix model [11] to synthesize a hurricane image from a density map. From the hurricane trajectory generated in the previous step, we first generate a hurricane density map using a Gaussian mixture. We then train conditional GANs that map the density map to a climate image whose channels are climate variables. The discriminator learns to classify between a fake and a real hurricane image, given a density map. The generator learns to fool the discriminator. In the pix2pix model, both the generator and discriminator take the input density map.

5 Experiments

We first evaluate our proposed models to track hurricanes on large-scale climate data. We also evaluate the performance improvements due to our proposed data augmentation. As our baselines, we compare our proposed model against Real-time MDNet [13], MDNet [24], ECO [5], and Siam-FC [2].

5.1 Experimental Settings

Dataset. We use 20-year-long records from 1996 to 2015 of *Community Atmospheric Model v5 (CAM5)* dataset. It contains snapshots of the global atmospheric states for every three hours. Each snapshot contains multiple physical variables, among which we use surface-level pressure (PSL), zonal wind (U850), and meridional wind (V850) given their relevance

↑ Density map
50 × 100 × 1
50 × 100 × 24
50 × 100 × 12
50 × 100 × 6
↑ Input

↑ Output
FC: flatten ×5
6 × 12 (64) (2 × 2)
12 × 25 (32) (2 × 2)
25 × 50 (24) (2 × 2)
50 × 100 (12) (2 × 2)
↑ Density map

Figure 3: (*Left*) 4-layer ConvLSTM architecture used for f, g. (*Right*) Bounding-box regressor (w) with 4-layer CNNs with a fully-connected layer; the numbers at each layer are the input size (the number of features) (the max-pooling window size).

to hurricane identification from scientific literature. As ground truth, we use the corresponding TECA labels [29], which contain the latitude and longitude of each hurricane and the diameter of hurricane-force winds.

In order to fit the model in memory, we split the global map into several patches within non-overlapping Northern tropical cyclone basins[2] of a $25^\circ \times 50^\circ$ sub-image ($1^\circ \approx 111$ *km*). From 400 hurricane trajectories in this dataset, we create 11,160 sub-sequences of ten frames (corresponding to 30 hours), and use 80% for training, 10% for validation, and the other 10% for testing. The input image size is 50×100 pixels with around 0.5° (55.5 *km*) resolution.

Models and Hyper-parameters. Both for the focus learning module and the tracking module, we use four-layered ConvLSTM architectures, as illustrated in Figure 3 (*left*). All the input-to-state and state-to-state kernels are of size 3×3 which is consistent with the size of hurricane. Both for the bounding box regressor and the discriminator, multi-layered CNNs followed by a fully-connected layer are utilized as in Figure 3 (*right*). The batch size is 24, and all ConvLSTM layers have the forget bias of 0.5. We use AdaGrad optimizer with the learning rate of 0.0005, and apply 10% dropout during training. We initialize all the states of the ConvLSTMs to zero before the first input comes, meaning that no information has been memorized in cell from the past. The Proposed tracking framework is built on Python using TensorFlow, trained and tested on a machine with Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz and six NVIDIA Tesla K80 with 12GB memory per GPU. The average testing speed of the proposed framework is about 809.32 images per second.

During training, we fix the video length to ten frames (30 hours). In testing, we divide the entire hurricane video (of an arbitrary length) for each 10 frames and test each sub-video separately. Except for the first sub-video, where the actual ground truth location is given, we give as input the estimated bounding box from the last frame of the previous sub-video in replacement of the initial location to the focus learning module. The density map input for the focus learning module has been synthesized based on the bounding box input on the fly; formally, $\mathcal{N}\left((x+\frac{w}{2}, y+\frac{h}{2}), \sigma^2\mathbf{I}\right)$, where the bounding box is given as $\{x, y, w, h\}$ and $\sigma = \max\{w, h\}$. To be consistent with the training phase, we initialize all hidden states of the ConvLSTM to zero.

Evaluation Metrics. We follow the evaluation protocol widely used in visual object tracking benchmarks [19, 40], using two evaluation criteria. First, a *success plot* shows the success ratio of each method for overlap thresholds from 0 to 1 (the x-axis of the plot), where we count a frame as success if the intersection over union (IOU) of the bounding boxes between prediction and ground truth is greater than the overlap threshold. Second, the *precision plot* shows the ratio of successful frames whose tracker output is within the given threshold (the x-axis of the plot, from 0 to 100 pixels) from the ground truth, measured by the center distance between bounding boxes. For both plots, we also report the area under

[2] Typically, seven commonly accepted basins include 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.

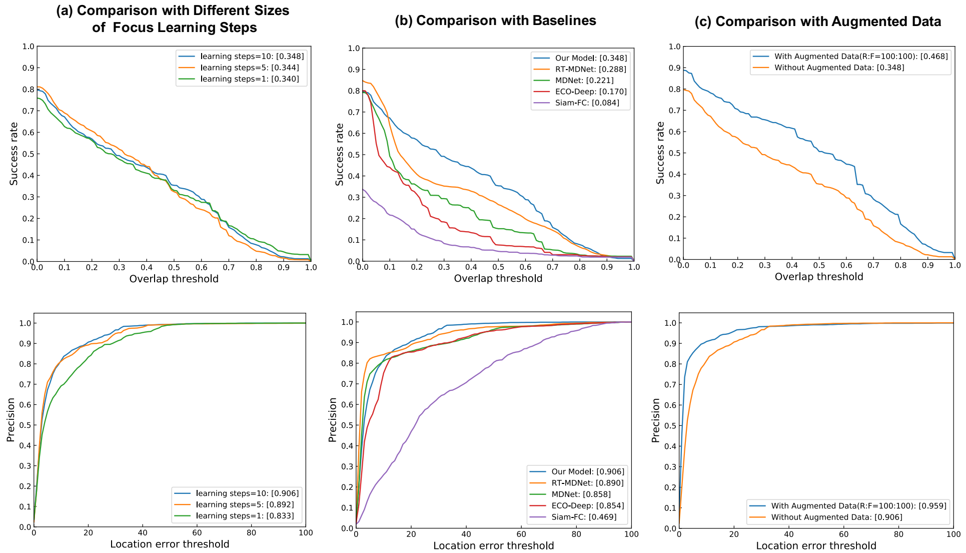

Figure 4: Success plot *(top)* and Precision plot *(bottom)* of experiments.

the curve (AUC) score of each method in the legend of the plot (higher values are better).

5.2 Hurricane Tracking Results

The main contribution of our tracking framework is the capability of the focus learning module to accurately learn the feature of the target by repeating learning steps. To show the effectiveness of repeated feature learning, we conduct comparison among variations of focus learning steps. Figure 4(a) shows the success plot and the precision plot of hurricane tracking trained and tested with CAM5 climate data. We observe that increasing focus learning steps slightly improves the tracking performance, because with larger learning steps, the model imprints the feature of target more strongly in its hidden state.

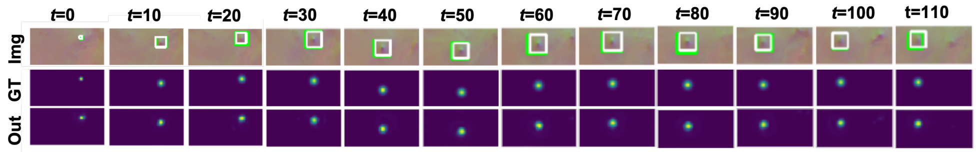

Figure 5: Hurricane tracking results. *Top*: inputs overlaid with ground truth (white) and prediction (green), *Middle*: ground truth density map, *Bottom*: predicted density map.

Figure 5 shows a challenging example of our hurricane tracking, where a new hurricane emerges from the right side of the image around $t = 90$. Both from the bounding box (top row) and density map (bottom row) results, one can see that our model robustly tracks the target hurricane from start to end for a long period ($t = 110$ in this example, corresponding to 330 hours) even at the presence of another similar object.

Lastly, we compare our tracking performance against other state-of-the-art baselines, including Real-time MDNet [13], MDNet [24], ECO [5], and Siam-FC [2]. We compare both success and precision plots including AUC scores. As shown in Figure 4(b), our tracking framework significantly outperforms all baselines. For instance, our proposed method outperforms the best performing baseline (RT-MDNET) by 20.8% in terms of the AUC of success rates.

5.3 Effects of Data Augmentation

To see the effect of data augmentation introduced in Section 4.3, we train the proposed model with the augmented data in conjunction with CAM5 data. Figure 6 illustrates an example of real CAM5 hurricane data in North America *(top)* and that of synthesized hurricane data when applying our data augmentation *(bottom)*. Visually inspecting the examples, we see that the synthesized hurricane data successfully mimics the key properties of hurricane, which has low pressure at the center and spirally rotating wind vectors with a counter-clockwise direction around the center.

As shown in Figure 4(c), data augmentation significantly improves the tracking performance. Comparing the performance with and without augmented data, we observe 34.5% improvement in the AUC of success rates over our best model without data augmentation.

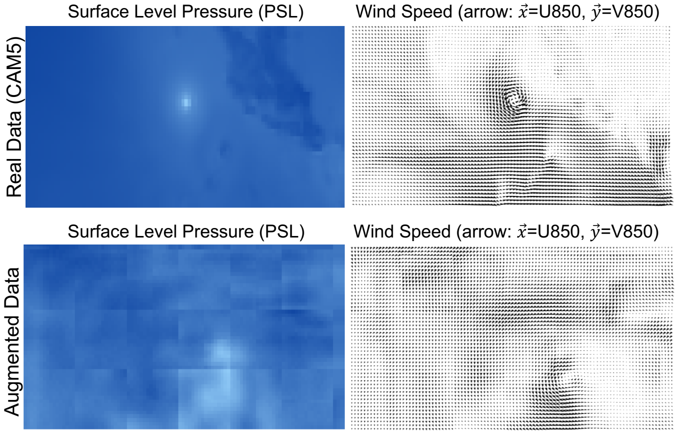

Figure 6: Real hurricane example (*top*) and augmented synthesized one (*bottom*) in North America for each climate variable channel.

6 Conclusions

This paper proposed a simple but robust end-to-end model to tackle the extreme climate event tracking problem. Due to its unique challenges, including wider ranges of spatio-temporal dynamics, the blur boundary of a target, and the shortage of labeled dataset, we design our ConvLSTM-variant model to learn a wide range of spatio-temporal dynamics even at the absence of the clear boundary of the target. Also, we presented a novel data augmentation approach based on conditional GANs to overcome the data shortage problem in climate science. Extensive experiments indicate that the proposed framework significantly improves hurricane tracking performance over several state-of-the-art methods.

Acknowledgement. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Lab. under contract DE-AC52-07NA27344(LLNL-CONF-776815). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (No. NRF-2018M3E3A1057305).

References

[1] Sheila Alemany, Jonathan Beltran, Adrian Perez, and Sam Ganzfried. Predicting hurricane trajectories using a recurrent neural network. In *ArXiv:1802.02548*, 2018.

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional Siamese networks for object tracking. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2016.

[3] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[4] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2016.

[5] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: efficient convolution operators for tracking. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] Lin Dong and Fuqing Zhang. OBEST: An observation-based ensemble subsetting technique for tropical cyclone track prediction. *Weather and Forecasting*, 31(1):57–70, 2016.

[7] Russell L. Elsberry, James R. Hughes, and Mark 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.

[8] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[9] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. A twofold siamese network for real-time object tracking. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017.

[12] Won-Dong Jang and Chang-Su Kim. Online video object segmentation via convolutional trident network. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[13] Ilchae Jung, Jeany Son, Mooyeol Baek, and Bohyung Han. Real-time MDNet. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2018.

[14] Sookyung Kim, Sasha Ames, Jiwoo Lee, Chengzhu Zhang, Aaron C. Wilson, and Dean Williams. Resolution reconstruction of climate data with pixel recursive model. In *Proc. of the IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 313–321, 2017.

[15] Sookyung Kim, Sasha Ames, Jiwoo Lee, Chengzhu Zhang, Aaron C Wilson, and Dean Williams. Massive scale deep learning for detecting extreme climate events. In *Climate Informatics*, 2017.

[16] Sookyung Kim, Hyojin Kim, Joonseok Lee, Sangwoong Yoon, Samira Ebrahimi Kahou, Karthik Kashinath, and Mr Prabhat. Deep-hurricane-tracker: Tracking and forecasting extreme climate events. In *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019.

[17] Sookyung Kim, Jungmin M Lee, Jiwoo Lee, and Jihoon Seo. Deep-dust: Predicting concentrations of fine dust in Seoul using LSTM. In *Climate Informatics*, 2019.

[18] Wonjik Kim and Osamu 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.

[19] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Luka Cehovin, Gustavo Fernandez, Tomas Vojir, Gustav Hager, Georg Nebehay, and Roman Pflugfelder. The visual object tracking VOT2015 challenge results. In *Proc. of the IEEE International Conference on Computer Vision (ICCV) workshops*, 2015.

[20] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[21] Yunjie Liu, Evan Racah, Mr Prabhat, Joaquin Correa, Amir Khosrowshahi, David Lavers, Kenneth Kunkel, Michael Wehner, and William Collins. Application of deep convolutional neural networks for detecting extreme weather in climate datasets. In *Proc. of the International Conference on Advances in Big Data Analytics*, 2016.

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

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

[24] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

[26] Liangbo Qi, Hui Yu, and Peiyan 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.

[27] Evan Racah, Christopher Beckham, Tegan Maharaj, Samira Ebrahimi Kahou, Prabhat, and Chris 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.

[28] Bernardino Romera-Paredes and Philip Hilaire Sean Torr. Recurrent instance segmentation. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2016.

[29] Oliver Rübel, Surendra Byna, Kesheng Wu, Fuyu Li, Michael Wehner, Wes Bethel, et al. TECA: A parallel toolkit for extreme climate analysis. *Procedia Computer Science*, 9:866–876, 2012.

[30] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[31] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

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

[33] Andrew D. Snyder, Zhaoxia Pu, and Yuejian 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.

[34] Jeany Son, Ilchae Jung, Kayoung Park, and Bohyung Han. Tracking-by-segmentation with online gradient boosting decision tree. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[35] Yilin Song, Chenge Li, and Yao Wang. Pixel-wise object tracking. *ArXiv:1711.07377*, 2017.

[36] Cong Thanh, Tran Tan Tien, and Kieu Quoc 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.

[37] Tran Tan Tien, Cong Thanh, Hoang Thanh Van, and Kieu Quoc Chanh. Two-dimensional retrieval of typhoon tracks from an ensemble of multimodel outputs. *Weather and Forecasting*, 27(2):451–461, 2012.

[38] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *ArXiv:1706.08033*, 2017.

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

[40] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[41] Feng Xiong, Xingjian Shi, and Dit-Yan Yeung. Spatiotemporal modeling for crowd counting in videos. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[42] Donghun Yeo, Jeany Son, Bohyung Han, and Joon Hee Han. Superpixel-based tracking-by-segmentation using markov chains. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.