A Deep Learning-Based Methodology for Precipitation Nowcasting With Radar

Lei Chen1, Yuan Cao2, Leiming Ma1, and Junping Zhang2

1Shanghai Central Meteorological Observatory, Shanghai, China, 2Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai, China

Abstract  Nowcasting and early warning of severe convective weather play crucial roles in heavy rainfall warning, flood mitigation, and water resource management. However, achieving effective temporal-spatial resolution nowcasting is a very challenging task owing to the complex dynamics and chaos. Recently, an increasing amount of research has focused on utilizing deep learning approaches for this task because of their powerful abilities in learning spatiotemporal feature representation in an end-to-end manner. In this paper, we present convolutional long short-term memory with a layer called star-shape bridge to transfer features across time steps. We build an end-to-end trainable model for the nowcasting problem using the radar echo data set. Furthermore, we propose a raining-oriented loss function inspired by the critical success index and utilize the group normalization technique to refine the convergence performance in optimizing our deep network. Experiments indicate that our model outperforms convolutional long short-term memory with the cross entropy loss function and the conventional extrapolation method.

1. Introduction

Precipitation nowcasting refers to providing precise and timely prediction of rainfall intensity over a local region in the near future. It is important in daily life because it can generate society-level emergency rainfall alerts and produce weather guidance for airports. Because of the high accuracy and spatiotemporal resolution requirements, precipitation nowcasting is quite challenging and has emerged as a hot research topic in the meteorological community (Sun et al., 2014).

Existing methods for precipitation nowcasting can roughly be categorized into two classes: numerical weather prediction (NWP)-based methods (Sun et al., 2014; Weisman et al., 2008) and radar echo extrapolation-based methods (Han et al., 2009; Li et al., 1995; Rinehart & Gravey, 1978). The NWP methods yield predictions by numerically solving partial differential equations subject to dynamic and thermodynamic laws. However, the inner mechanisms that underpin atmospheric behavior, which is regarded as a complex spatiotemporal dynamical system, are not fully understood. Hence, NWP-based methods for precipitation nowcasting have essential limitations. Meanwhile, some conventional extrapolation methods, such as COTREC (Mecklenburg et al., 2000; Rinehart & Gravey, 1978), adopt computer vision techniques for precipitation nowcasting. Although the extrapolation methods are effective for reliably predicting radar echo, they lack a strong nonlinear mapping ability to extract inherent features from large amounts of data.

To address these issues, deep learning-based methods utilize large-scale historical data to learn a collection of feature representations for precise prediction (Ranzato et al., 2014; Shi et al., 2015; Zheng et al., 2015). By formulating precipitation nowcasting as a spatiotemporal sequence forecasting problem, that is, predicting the next consecutive frame sequence subsequent to the previous observation sequence (Shi et al., 2015), the pioneering long short-term memory (LSTM) framework proposed in (Sutskever et al., 2014) provides a general deep learning framework for sequence-to-sequence predictive problems by concatenating two LSTMs: one for the input sequence and another for the output sequence. However, the LSTM predictive learning
methods focus more on modeling temporal variations, such as object motion. To effectively model spatiotemporal information, Shi et al. (2015) extended the LSTM structure to the convolutional long short-term memory (ConvLSTM) model by exploiting convolutions in both input-to-state and state-to-output transitions. Despite their seminal effort, there exist some deficiencies. Although better than the conventional extrapolation methods used in capturing spatiotemporal correlations, first, the ConvLSTM can be further optimized. Second, their model is optimized by assuming cross entropy loss as a reasonable proxy index of critical success index (CSI), which is widely used in precipitation nowcasting. Finally, the radar echo images used in their work have 100 × 100 pixels, which may not be representative of the dimensionalities in operational forecast systems. When working with high resolution radar echo data sets of 500 × 500 pixels in our study, inevitable small batches and limited GPU memory will degrade the convergence performance of the model.

In this paper, we aim to address these three problems. We introduce a new ConvLSTM with star-shaped bridge architecture to transfer information across time steps (Cao et al., 2019). We also discuss group normalization (GN), which improves the performance of the optimization process. Inspired by the CSI, we employ a special multisigmoid loss that is regarded as the differentiable CSI. Our loss function is suitable for optimization based on stochastic gradient descent.

We compare the results to other baselines, including COTREC and the ConvLSTM with cross entropy as the loss function. It is shown that we achieve considerable performance improvement in comparison with the other models.

### 2. Data

The radar echo data set employed in this paper includes composite reflectivity data collected from the dual polarization Weather Surveillance Radar-1988 Doppler Radar (WSR-88D) located in Pudong, Shanghai, as indicated in Figure 1. The data set contains 170,000 echo images generated by volume scans in intervals...
of approximately 6 min from October 2015 to July 2018. The echo data are interpolated to 0.01° × 0.01° longitude-latitude grids from the radar polar coordinates. To reduce the noise, we further remove the pixels with correlation coefficients of less than 0.85 (Kumjian et al., 2010). We also filter out some nonconsecutive frames due to radar faults and maintenance and apply a stride-1 sliding window to generate 76,779 length-20 sequence samples.

The training set and the testing set contain 44,099 and 32,700 sequence samples, respectively. We rescale the logarithmic reflectivity $R$ to the interval $[0, 1]$ by setting $\hat{R} = \frac{R}{70}$.

### 3. Methods

#### 3.1. COTREC

COTREC, the continuous TREC (tracking radar echo by correlation) method, has been well described in several papers (Li et al., 1995; Mecklenburg et al., 2000; NOVAK, 2007). Due to space considerations, here, we only provide a brief review of the methodology. The TREC method is applied to two successive radar echo images. The first and second images (at times $T_1$ and $T_2 = T_1 + \Delta t$) are divided into a number of equally sized boxes as two-dimensional pixel arrays. The arrays of reflectivity in the first image will then be cross correlated with the arrays in the second image. The correlation coefficient $R$ is used as an objective criterion for the agreement between the patterns of investigated arrays and is calculated using the following formula:

$$R = \frac{\sum_{i=1}^{N} Z_1(i) \times Z_2(i) - \frac{1}{N} \sum_{i=1}^{N} Z_1(i) \sum_{i=1}^{N} Z_2(i)}{\left( \sum_{i=1}^{N} Z_1^2(i) - N \bar{Z}_1^2 \right) \times \left( \sum_{i=1}^{N} Z_2^2(i) - N \bar{Z}_2^2 \right)}$$

(1)

where $Z_1$ and $Z_2$ are the pixel arrays of reflectivity at $T_1$ and $T_2$, respectively. The summation is carried out for all pixels in the array, and $N$ is the total number of pixels in the corresponding array. The computation can be repeated for all possible arrays of the second image within a search area centered on the arrays of the first image. The center of the array of the second image with the highest correlation will represent the endpoints of motion vectors (i.e., TREC vectors).

Before the whole vector field is used, each motion vector is corrected for consistency. A three-step procedure is applied:

1. Vectors with zero velocity are replaced by average vectors of the neighboring vectors. If a vector deviates by more than 25° from the local mean of its neighborhood, then it is also replaced by the mean vector.
2. To produce the gridded vector field, a two-pass objective analysis is conducted with the Cressman weighting function.
3. The third step uses a variational technique (COTREC) (Bertsekas, 1982), forcing the vector field to zero divergence when close to the original vector field. The variational analysis can be expressed as follows:

$$\begin{align*}
(\hat{u}, \hat{v}) &= \arg \min_{(u,v)} \int \int_D (u - u^o)^2 + (v - v^o)^2 \, dx \, dy \\
& \text{s.t. } \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0
\end{align*}$$

(2)

where $u^o$ and $v^o$ are the $x$ and $y$ components of the original vectors, and $u$ and $v$ are their corresponding solutions.

Temporal extrapolation makes use of the semi-Lagrangian scheme (Germann & Zawadzki, 2001). Obtaining the patterns of echo growth or decay is straightforward: The difference in the echo intensity along the flow between two successive radar images is deduced by linear extrapolation. Additionally, to prevent unrealistic growth, growth will be terminated after three prediction steps or when the tracked patch is stronger than 50 dBZ. In the case of decay, the extrapolation will also be stopped after three prediction steps or when the patch is weaker than 20 dBZ. Although the process of intensity extrapolation is empirical, we realize some improvement relative to COTREC by focusing only on movement.

#### 3.2. Group Normalization

Batch normalization (BN) (Ioffe & Szegedy, 2015) plays an important role in deep learning. BN computes the mean and variance for the samples of one batch to normalize feature maps. Despite its great success, BN
Figure 2. Schematic representation of group normalization.

exhibits drawbacks that are also caused by its distinct behavior of normalizing along the batch dimension. A small batch size leads to inaccurate estimation of batch statistics. This limits BN’s usage for dealing with high resolution images (e.g., radar echo images), which requires small batches constrained by GPU memory consumption.

Recently, Wu and He (2019) proposed a novel form of normalization named group normalization (GN). Distinct from BN, GN normalizes features by grouping feature channels into small groups and computes the mean and variance for each small group. In this way, GN can normalize the feature tensors within each single sample in the batch. As a result, the normalization can be independent of the batch size. For the task of precipitation nowcasting using radar echo images, we discover the effectiveness of using GN as the normalization layer. We give a brief description about GN in the following.

In the case of 2D images, \( i = (i_N, i_C, i_H, i_W) \) is the index of 4D tensor features in \((N, C, H, W)\) order, where \( N \) is the batch axis, \( C \) is the channel axis, and \( H \times W \) are the numbers of grids in the height and width directions (Figure 2). GN performs the following computation:

\[
\begin{align*}
y_i &= \gamma \hat{x}_i + \beta \\
\hat{x}_i &= \frac{1}{\sigma_t} (x_i - \mu_t) \\
\mu_t &= \frac{1}{m} \sum_{k \in S_t} x_k, \quad \sigma_t = \sqrt{\frac{1}{m} \sum_{k \in S_t} (x_k - \mu_t)^2} \\
S_t &= \left\{ \ldots r \ldots I \ldots | r_N = i_N, \left\lfloor \frac{r_C}{q} \right\rfloor = \left\lfloor \frac{i_C}{q} \right\rfloor \right\}, \quad q = \frac{C}{G}
\end{align*}
\]

where \( \gamma \) and \( \beta \) are trainable scale and shift, \( x \) is the feature computed by a layer, and \( i \) is the index. \( C \) is the number of channels, \( G \) is the number of groups and \( q \) is the number of channels per group. \( \lfloor \cdot \rfloor \) is the floor operation, which means that \( r \) and \( t \) are in the same group if their indices along the \( C \) axis are equal in the sense of division.

3.3. ConvLSTM With Star-Shaped Bridge

Suppose that we observe a dynamical system of \( K \) measurements (e.g., radar echo images) in a time (e.g., radar scan cycle) over a spatial region with an \( M \times N \) grid. Thus, the observation of these \( K \) measurements at any time can be represented by a tensor \( \mathcal{X} \in \mathbb{R}^{K \times M \times N} \). Define the \( K \)-length observation as \( \mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_K \). The spatiotemporal predictive learning problem is to predict the most likely \( L \)-length sequence in the future given the previous \( K \)-length observation. The goal can be formulated as follows:

\[
\hat{\mathcal{X}}_{t+1}, \ldots, \hat{\mathcal{X}}_{t+L} = \arg \max_{\mathcal{X}_{t+1}, \ldots, \mathcal{X}_{t+L}} p(\mathcal{X}_{t+1}, \ldots, \mathcal{X}_{t+L} | \mathcal{X}_{t-K+1}, \ldots, \mathcal{X}_t)
\]

(4)

Compared with the standard LSTM, ConvLSTM (Shi et al., 2015) is able to simultaneously model the spatiotemporal structures by encoding the spatial information into tensors. In ConvLSTM, all inputs \( \mathcal{X}_t \) (\( t \in [1, \ldots, K] \)), cell outputs \( C_t \), hidden states \( H_t \), and gates \( i_t, f_t, g_t \), and \( o_t \) are 3-D tensors. ConvLSTM determines
the future state of a certain grid according to the inputs and past states of its local neighbors in the encoder-decoder architecture (see Figure 3):

\[
\begin{align*}
g_t & = \tanh(W_{xg} \ast x_t + W_{rg} \ast h_{t-1} + b_g) \\
i_t & = \sigma(W_{xi} \ast x_t + W_{hi} \ast h_{t-1} + W_{ci} \ast c_{t-1} + b_i) \\
f_t & = \sigma(W_{xf} \ast x_t + W_{hf} \ast h_{t-1} + W_{cf} \ast c_{t-1} + b_f) \\
c_t & = f_t \odot c_{t-1} + i_t \odot g_t \\
o_t & = \sigma(W_{xo} \ast x_t + W_{ho} \ast h_{t-1} + W_{co} \ast c_{t-1} + b_o) \\
h_t & = o_t \odot \tanh(c_t)
\end{align*}
\]

where \( \sigma = \frac{1}{1 + \exp(-x)} \) is the sigmoid activation function, and \( \ast \) and \( \odot \) denote the convolution operator and the Hadamard product, respectively. If the states are viewed as hidden representations of moving objects, then ConvLSTM with a larger transitional kernel should be able to capture faster motions, while ConvLSTM with a smaller kernel can capture slower motions (Shi et al., 2015). The input gate \( i_t \), forget gate \( f_t \), output gate \( o_t \), and input modulation gate \( g_t \) control information flow across the memory cell \( c_t \).

In the conventional ConvLSTM architecture, as illustrated in Figure 3, the information is conveyed only upward and is updated horizontally. In our nowcasting task, the information from representations extracted at previous different level convolutional layers may be useful. In this paper, we introduce a novel transition...
| Model                        | CSI20 (30 min) | CSI20 (60 min) |
|-----------------------------|----------------|----------------|
| COTREC                      | 0.449          | 0.341          |
| ConvLSTM + CE               | 0.453 (0.89%)↑ | 0.363 (6.45%)↑ |
| ConvLSTM + CE + GN          | 0.554 (23.4%)↑ | 0.457 (34.0%)↑ |
| ConvLSTM + SSL10 + GN       | 0.574 (27.8%)↑ | 0.469 (37.5%)↑ |
| ConvLSTM + SSL20 + GN       | **0.600 (33.6%)↑** | **0.479 (40.5%)↑** |
| ConvLSTM + SSL30 + GN       | 0.569 (26.7%)↓ | 0.470 (37.8%)↓ |

Note. For each setup of ConvLSTM, we also provide the performance improvement percentage relative to COTREC. The bold emphasis indicates the configuration of best performance.

path named star-shaped bridge (SB) (Cao et al., 2019). SB adds more information from the last time step to make the feature flow in multilayer ConvLSTM more robust. Concretely, we concatenate outputs of all ConvLSTM layers and pass them to a convolution layer with kernel size $1 \times 1$ and then split the output to all ConvLSTM layers of the next time step using a residual connection (Figure 4). Throughout this paper, ConvLSTM with SB will be referred to as ConvLSTM for clarity.

### 3.4. Multisigmoid Loss

The CSI, which describes the ability to predict the occurrence above a given threshold (e.g., 20 dBZ), is widely applied in the meteorological community. CSI is defined as $\text{CSI} = \frac{TP}{TP+FN+FP}$. For each pixel, we define terms as follows: $TP = \text{hits}$, $FN = \text{misses}$, and $FP = \text{false alarms}$. However, CSI is not a suitable loss function for gradient descent optimization due to its nondifferentiability. Shi et al. (2015) adopted cross entropy loss ($CE$), defined as

$$CE = \sum_k I_k \log \hat{I}_k + (1 - I_k) \log(1 - \hat{I}_k)$$  \hspace{1cm} (6)

where $\hat{I}_k$ is the scaled echo intensity of the observation at the pixel indexed $k$ and $I_k$ is the predicted echo intensity. However, $CE$ can only be regarded as a proxy index of CSI because $I_k$ and $\hat{I}_k$ are not true probabilities, though the values are between 0 and 1. To overcome this drawback, we employ sigmoid loss $L_i$ to simulate CSI at threshold $i$:

$$L_i^c = \sum_k (\sigma(I_k - i) * c) - \sigma((\hat{I}_k - i) * c))^2$$  \hspace{1cm} (7)

where $c$ is the scale factor, a hyperparameter used to control the slope of the sigmoid function $\sigma$. In this work, we adopt $i \in \{20, 30\}$. Further, the multisigmoid loss is defined as

$$SSL^c = L_{20}^c + L_{30}^c$$  \hspace{1cm} (8)

### 4. Experiment

In this section, we evaluate the performance of our model with respect to the radar echo data set. We also compare our model with some baselines, including COTREC and ConvLSTM with $CE$ loss.

#### 4.1. Setup of the Experiment

In the case of COTREC, the box arrays are set to have a fixed size of $19 \times 19$ pixels, the centers of these pixel arrays are spaced 5 pixels apart, and the entire COTREC vector field has a resolution of $99 \times 99$. The search radius is 11 pixels.

All of our deep learning-based experiments are implemented with PyTorch and conducted on four NVIDIA GTX 1080Ti GPUs. We train all models with the ADMA optimizer, and the learning rate is 0.002. The batch size is set to 4, and we employ optimization iterations of up to 60 epochs at most. Our network includes two-layer ConvLSTM, with each layer containing 128 hidden states and a $3 \times 3$ filter size. For GN, we set $C$ to 128 and $G$ to 4 for all convolution layers. We use the training set as input during parameter adjustment and
evaluate the performance with respect to the test data set every epoch. The training process is terminated according to the stop learning early stopping strategy.

Under the condition of the deep learning-based method, the contributions of GN and multisigmoid loss are investigated step-by-step. Additionally, we test the influence of different scale factors $c$ of multisigmoid loss for robustness.

### 4.2. Results

All results are shown in Tables 1 and 2 and Figure 5. We find that the ConvLSTM without GN achieves some performance improvement. However, the average CSI30 over 30 min falls by 7.05%, which may be caused by statistical instability of batch estimation. When working with BN, ConvLSTM significantly outperforms COTREC for all investigated indices. The ConvLSTM with GN provides 24.5% and 33.1% improvements over the average CSI20 predicted during 30 and 60 min. It also raises the CSI30 (30 min) and CSI30 (60 min) by 9.29% and 33.5%. For investigating the influence of different scale factor $c$ values in SSL loss, we carry out three experiments with $SSL_{10}$, $SSL_{20}$ and $SSL_{30}$ as loss functions. As shown in Tables 1 and 2, $c = 10$ and $c = 20$ are appropriate, while 20 is the best choice. $c = 30$ is not as effective for our task because of the limited performance for CSI30. The performance degradation on the large scale could be attributed to the resultant gradient, which is too steep to search for a stable descent direction. The corresponding frame-by-frame quantitative comparisons are also presented in Figure 5. It can be seen from Figure 5 that our model produces the best result over all time steps. On the other hand, the contribution of GN is most obviously shown by the comparison between ConvLSTM$+CE$ and ConvLSTM$+CE$+GN. In Figure 5, the results with the 30 dBZ threshold, COTREC is slightly better than ConvLSTM$+CE$ and ConvLSTM$+CE$+GN at the first time step. The most probable explanation for this is that COTREC tries to include all details at small scales, which typically have short time step lifetimes and rapidly decorrelate, while effective prediction for small scale features with a long leading time is beyond the information capacity supplied by only echo images. Our model learned to focus on features at larger scales while ignoring the small scales, and this learned strategy slightly sacrifices the accuracy in the first one or two time steps to achieve better overall performance.

![Figure 5](image-url). Frame-by-frame comparison of CSI with thresholds of 20 dBZ (a) and 30 dBZ (b) based on COTREC and ConvLSTM networks. For clarity, we only show the result of scale factor $c = 20$, the best choice.
Figure 6. Prediction example started at 01:00 UTC on 13 July 2019. The right column shows predicted patterns at 30 min (01:30), and the left column shows those at 60 min (02:00). Panels (a) and (b) are observations, (c) and (d) are predictions of our model, and (e) and (f) are predictions of COTREC. The areas in the three boxes are investigated.

For comparison, a recent predicted case of COTREC and our model is shown in Figure 6. The three areas in the boxes are investigated. In Area1, our model achieves more reasonable intensity prediction and can more accurately infer the contour, especially at the boundary for the 60 min prediction. For Area2, COTREC gives a prediction with a northward bias and maintains the intensity at 60 min. Our model provides more accurate direction forecasting and underestimates the intensity in Area2. However, the level of underestimation is acceptable from the perspective of misses with the threshold of 30 dBZ. For Area3, notable differences exist. As for area1, our model offers reliable shape prediction and acceptable underestimated intensity with regard to the threshold of 30 dBZ. It is obvious that COTREC incorrectly estimates the echo movement speed.
and shape compared to the observation. The limited performance of COTREC in Area1 and Area3 leads to serious misses and false alarms, which ultimately degrade the overall prediction performance. Note that the blurring effect of our model may be caused by the inherent uncertainties of the task. For long-term prediction, our model tends to blur the predictions to alleviate the error after the training process. Although the deep learning model performs better than COTREC for this case, it is worth noting that radar echoes during 2019 are not included in the test data set, and a robust conclusion should be drawn from a comparison conducted on the data set of 2019 in future work.

5. Summary and Discussion

In this paper, we introduce a novel ConvLSTM network with the star-shaped bridge layer for passing information. Information can be transferred both horizontally and vertically. Owing to the nondifferentiable nature of CSI, which prevents it from serving as a loss function for the gradient descent method, we employ a special multisigmoid loss function as a simulation of CSI. Additionally, we discuss the benefits of group normalization. The result shows that our model achieves state-of-the-art performance. The contributions of this paper are stated as follows:

1. We present the ConvLSTM with star-shaped bridge for precipitation nowcasting.
2. We introduce the GN technology to refine the convergence performance in optimization for ConvLSTM.
3. We employ a special loss function that is more in line with CSI.

It should be noted that GN offers the best performance improvement relative to COTREC due to its capability of stably estimating batch statistics; meanwhile, the special loss function is responsible for the improvement to some extent. Although COTREC achieves better performance with respect to intensity in some specific areas, the obvious deviations in the direction and shape of the forecast degrade the overall performance. Our model achieves a more reliable forecast with an acceptable underestimation level in terms of a proper threshold.

It should be noted that our model suffers from a blurring effect, especially for longer future time steps. One possible interpretation is that there exists an intrinsic predictability property of nonlinear systems using the available data sources. The more blurred the forecasted images are, the more the uncertainty of the system will grow with increasing lead time. To further leverage the ability of the complex deep learning-based model to obtain sharper and more accurate predictions, combining it with physical knowledge from the numerical weather forecast model and utilizing multimodal data, such as satellite data and ground weather observations, merit further exploration.

Acknowledgments

This work is supported by National Natural Science Foundation of China (41975069), Science and Technology Commission of Shanghai Municipality (18DZ1200400), the National Key Research and Development Project of China (2016YFC0201900) and ZJLab. We would like to thank Xingjian Shi for his excellent technical support and valuable discussion. Our data set is available online (at https://doi.org/10.7910/DVN/2GKMQJ).

References

Bertsekas, D. P. (1982). Constrained optimization and lagrange multiplier methods. New York: Academic Press.
Cao, Y., Li, Q., Chen, L., Zhang, J., & Ma, L. (2019). Video prediction for precipitation nowcasting. arXiv preprint arXiv:1907.08069.
Germann, U., & Zawadzki, I. (2001). Scale-dependence of the predictability of precipitation from continental radar images. Part I: Description of the methodology. Monthly Weather Review, 130(12), 2859–2873.
Han, L., Fu, S., Zhao, L., Zheng, Y., Wang, H., & Lin, Y. (2009). 3D convective storm identification, tracking, and forecasting: an enhanced titan algorithm. Journal of Atmospheric and Oceanic Technology, 26(4), 719–732.
Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning.
Kumjian, M. R., Ryzhkov, A. V., Melnikov, V. M., & Schuur, T. J. (2010). Rapid-scan super-resolution observations of a cyclic supercell with a dual-polarization wsr-88d. Monthly Weather Review, 138(10), 3762–3786.
Li, L., Schmid, W., & Joss, J. (1995). Nowcasting of motion and growth of precipitation with radar over a complex orography. Journal of Applied Meteorology, 34(6), 1286–1300.
Mecklenburg, S., Joss, J., & Schmid, W. (2000). Improving the nowcasting of precipitation in an alpine region with an enhanced radar echo tracking algorithm. Journal of Hydrology, 239(1-4), 46–68.
Novak, P. (2007). The czech hydrometeorological institute’s severe storm nowcasting system. Atmospheric Research, 83(2), 450–457.
Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., & Chopra, S. (2014). Video (language) modeling: A baseline for generative models of natural videos. arXiv preprint arXiv:1412.6604.
Rinehart, R. E., & Gravey, E. T. (1978). Three-dimensional storm motion detected by convective weather radar. Nature, 273, 287–289.
Shi, X., Chen, Z., Wang, H., Dit Yan, Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Advances in neural information processing systems, pp. 802–810.
Sun, J., Xue, M., Wilson, J. W., Zawadzki, I., Ballard, S. P., Onvlee-Hooimeyer, J., et al. (2014). Use of NWP for nowcasting convective precipitation: Recent progress and challenges. Bulletin of the American Meteorological Society, 95(3), 409–426.
Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems, pp. 3104–3112.
Weisman, M. L., Davis, C., Wang, W., Manning, K. W., & Klemp, J. B. (2008). Experiences with 0–36-h explicit convective forecasts with the WRF-ARW model. Weather and Forecasting, 23(3), 407–437.
Wu, Y., & He, K. (2019). Group Normalization. *International Journal of Computer Vision*. https://doi.org/10.1007/s11263-019-01198-w

Zheng, Y., Yi, X., Li, M., Li, R., Shan, Z., Chang, E., & Li, T. (2015). Forecasting fine-grained air quality based on big data. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, ACM, pp. 2267–2276.