water

MDPI

Article

Gated Attention Recurrent Neural Network: A Deeping Learning Approach for Radar-Based Precipitation Nowcasting

Guanchen Wu [1], Wenhui Chen [2] and Hoekyung Jung [3,*]

[1] Department of Information Engineering, Guizhou Communications Polytechnic, Guiyang 551400, China
[2] College of Computer Science and Technology, Hengyang Normal University, Hengyang 421002, China
[3] Department of Computer Science and Engineering, Paichai University, 155-40 Baejae-ro, Daejeon 35345, Korea
* Correspondence: hkjung@pcu.ac.kr; Tel.: +82-425205640

Abstract: Precipitation nowcasting predicts the future rainfall intensity in local areas in a brief time that impacts directly on human life. In this paper, we express the precipitation nowcasting as a spatiotemporal sequence prediction problem. Predictive learning for a spatiotemporal sequence aims to construct a model of natural spatiotemporal processes to predict the future frames based on historical frames. The spatiotemporal process is an abstraction of some of the spatial things in nature that change with time, and they usually do not change very dramatically. To simplify the model and facilitate the training, we considered that the spatiotemporal process satisfies the generalized Markov properties. The natural spatiotemporal processes are nonlinear and non-stationary in many aspects. The processes are not satisfied with the first-order Markov properties when making predictions, such as the nonlinear movement, expansion, dissipation, and intensity enhancement of echoes. To describe such complex spatiotemporal variations, higher-order Markov models need to be used for the modeling. However, many of the previous models for spatiotemporal prediction constructed were based on first-order Markov properties, losing information on the higher-order variations. Thus, we propose a recurrent neural network which satisfies the multi-order Markov properties to create more accurate spatiotemporal predictions. In this network, the core component is the memory cell structure of the gated attention mechanism, which combines the current input information, extracts the historical state that best matches the existing input from the historical multi-period memory information, and then predicts the future. Through this principle of the gated attention, we could extract the historical state information that is richer and deeper to predict the future and more accurately describe the changing characteristics of motion. The experiments show that our GARNN network captures the spatiotemporal characteristics better and obtains excellent results in the precipitation forecasting with radar echoes.

Keywords: deep learning; recurrent neural network; spatiotemporal prediction; precipitation nowcasting

Citation: Wu, G.; Chen, W.; Jung, H. Gated Attention Recurrent Neural Network: A Deeping Learning Approach for Radar-Based Precipitation Nowcasting. *Water* **2022**, *14*, 2570. https://doi.org/10.3390/w14162570

Academic Editor: Ataur Rahman

Received: 29 July 2022
Accepted: 19 August 2022
Published: 20 August 2022

1. Introduction

Spatiotemporal predictive learning is an essential branch of predictive learning, and it has rich potential application scenarios for many practical problems, such as precipitation nowcasting [1–4], traffic flow prediction [5,6], behavior recognition prediction [7], physical scene understanding [8], and video understanding [9]. Such a wide range of potential applications have attracted increasing interest in the machine learning and deep learning communities. Meanwhile, many fruitful methods have also been proposed. Among these methods, to fully capture the relationship of spatiotemporal data's dependencies in both the temporal and spatial dimensions, t a Recurrent Neural Network (RNN) has been adopted, with stacked Convolutional Long Short-Term Memory (ConvLSTM) [10] units. This recurrent neural network is mainly inspired by the widely used natural language-processing-related technologies, such as machine translation, audio recognition, and natural

language understanding. These scenarios have rich and skillful means of mining and using sequence data time-series information.

One of the tricks is that the ConvLSTM unit includes a memory unit, which stores the historical variations. Some of the studies have shown that the information retained by the memory unit is mainly the hidden state's gradient of the ConvLSTM. On the one hand, the cumulative retention of the information of the hidden state's gradient makes the historical information more effective for backward transmission; on the other hand, it significantly alleviates the RNN gradient's disappearance problem. Another trick is that the ConvLSTM unit contains a series of gated structures. These gated structures are mainly based on the current input information, enabling adaptive transfer control and selective access to the historical information.

There are other studies, in addition to storing the historical variations in the time dimension to describe the changes in the time series, and adding the ConvLSTM unit to store the spatial detail variations in the spatial dimension. However, when the ConvLSTM unit models the spatiotemporal prediction model, its internal state transition may not be optimal. Due to the changes in the spatiotemporal dimension of the natural scene being full of randomness, the whole process could be highly nonlinear and non-stationary. Although it can be considered to decompose the non-stationary process into the sum of a deterministic time-varying polynomial and a zero-mean random variable, according to the Cramer decomposition, some high-order variation information will still be lost in the actual model learning.

Firstly, the convolution parameters in the ConvLSTM unit are highly shared in the spatiotemporal dimension, which leads to the penalty of spatiotemporal prediction learning mainly for the overall changes in the spatiotemporal process. They cannot regard the penalties for some of the local and highly nonlinear changes. The main reason for sharing these parameters in the spatiotemporal dimension is the graphics card's memory limitation. Secondly, the status transition in the ConvLSTM unit only relies on the status information of the previous moment. It is based on the assumption that the spatiotemporal process described approximates a first-order Markov process or a first-order Markov process converted by some transformation. For some of the slow and general nonlinear spatiotemporal processes, this assumption is reasonable. However, for a spatiotemporal process, such as radar echo, due to the sudden appearance and expansion, shrinking, disappearing, and other highly nonlinear changes that are frequently occurring, many of the variations will be lost based on this assumption. Finally, the gate structure is adopted in the ConvLSTM unit. The characteristic of the gated structure is that each transfer of historical status information is a selective transfer after synthesizing the current input information. To some extent, this selective transfer is beneficial to the current moment, there is no guarantee that it is also beneficial to the status information transfer in the next moment.

Therefore, we consider, in the time dimension, the ConvLSTM unit should add more historical state information, that is, not relying on only the state of the previous moment, but also relying on the state of the earlier moments. Combining our knowledge of the ConvLSTM unit, we propose a spatiotemporal predictive recurrent neural network, with the gated attention mechanism (GARNN). The network inherits the commonly used stacked structure, meaning the mode of stacked cell blocks. The design of the cell block refers to the Spatiotemporal Long Short-Term Memory (ST-LSTM) structure in the Predictive Recurrent Neural Network (PredRNN) model [11], and the memory flow in the spatial dimension, which is added to improve the ability of the cell block to maintain the small shape changes in the spatial dimension. The cell block achieves innovation in the time dimension by relying on the state information of multiple moments in the past. When we obtain the state information of multiple moments in the past, we do not simply accumulate and sum the past state information after transforming it, but refer to the attention mechanism. Because the states of the multiple historical moments have different influences on the future, to effectively utilize the state information with great influence, the attention mechanism is used to selectively extract the information of the past moments. Based on the current input

information, the meaningful historical state information is selectively obtained from the state information of the multiple historical moments. Then, the meaningful historical state information at each moment is synthesized to obtain the most valuable information for the current input.

Intuitively, the spatiotemporal prediction recurrent neural network, GARNN, of the gated attention mechanism that we propose, no longer passively accepts the historical state of the previous moment but actively selects the historical status of multiple moments. This network construction provides structural support for the highly nonlinear changes, making it possible for the neural networks to learn them. We trained and validated the GARNN model on the radar echo prediction dataset. Compared with the PredRNN model and the ConvLSTM model, the prediction performance of our proposed recurrent neural network with the gated attention mechanism was greatly improved.

2. Related Work

According to the different neural network structures, the spatiotemporal process's prediction learning can be roughly divided into three types [12,13]: 1. Generative Adversarial Network (GAN)-based methods; 2. RNN-based methods; 3. Convolutional Neural Network (CNN) and RNN-based methods.

Mathieu et al. [14] proposed a multi-scale GAN architecture model based on the differential loss function of the image gradient, which solves the problem of predicting the video frame blur to a certain extent. Liu et al. [15] added the spatial and motion constraint losses on top of the loss based on the image intensity and gradient and then used a FlowNet network to compute the optical flow information to predict temporally consistent frames. Similarly, Yi et al. [16] proposed a GAN network structure with a dual learning mode, using the relationship between the multiple domains for the image-to-image translation. Liang et al. [17] proposed a Dual Motion GAN (DMGAN) architecture that ensures that the model-predicted future frames are consistent with the pixel flow in the video, through a dual learning mechanism. The dual training method ensures that the predicted optical flow can help the network to reason, making the predicted future frames more realistic, and the future frame prediction task also makes the predicted optical flow information more realistic. Kwon et al. [18] proposed a unified generative adversarial network (comprising a generator and two discriminators) to predict the video frames accurately, maintain the consistency of the predicted past and future frames with the video sequence through circular retrospective restrictions, and reduce the blurring of predicted frames. Compared with the RNN models, these models transform complex state transitions into the operations between convolutional channels by stacking convolutional layers, so they are often unable to effectively capture the dependencies between video frames that are widely separated in time.

The RNN model was initially proposed to process the one-dimensional time-series information, such as text, which better captures the correlation of sequence elements in a time series. Given this perspective, some scholars tried to use the RNN model to predict future sequences according to the historical video sequences [19–21]. Ranzato et al. [22] proposed a recurrent convolutional neural network inspired by language modeling. Under the assumption of local space and time stationarity, the visual features generated by the clustering image patches were used to predict the future video frames. Srivastava et al. [23] used an LSTM Encoder to map the input video sequence to a fixed-length representation, and then used a single-layer/multi-layer LSTM Decoder to decode the learned representation, reconstruct the input video sequence and predict the future video sequences. Babaeizadeh et al. [24] believed that the future of many natural processes is not deterministic, and there may be multiple reasonable futures, so they proposed a Stochastic Variational Video Prediction (SV2P) method that predicts a different probable future for each sample of its potential variables. Denton et al. [25] introduced an unsupervised video generation model. The model learned a prior model of the uncertainty in the given environment and then drew samples from this prior model and combined them with the

deterministic estimation of future frames to generate the final future frames. Some other scholars also believed that the future is not certain and have completed a lot of preliminary research [26–30]. These RNN-based methods mainly model the time series relationship and characterize the uncertainty. However, they mainly model the spatiotemporal process's high-level features dynamically, leading to the inevitable loss of detailed information in the actual natural process.

To address the loss of detailed information in the RNN model when modeling the spatiotemporal processes, Shi et al. [10] combined the advantages of the convolution operation and LSTM. They proposed the convolutional neural network ConvLSTM that used convolution instead of the original matrix multiplication operation, allowing LSTM to maintain the two-dimensional characteristics of the image. Shi et al. [31] also combined the convolution operation with the GRU model [32,33], and proposed the Trajectory GRU (TrajGRU) model, which can actively learn the position-changing structure of the recurrent connections. Wang et al. [34] proposed an Eidetic 3D LSTM (E3D-LSTM), which integrated 3D convolution into RNN, and the encapsulated 3D-Conv enabled the local perceptron of RNN to have motion-sensing capability and enabled the memory cells to store better short-term features. Wang et al. [35,36] proposed a PredRNN model using zigzag memory flow, the core of which is a new spatiotemporal LSTM (ST-LSTM) unit, which can simultaneously extract and memorize variations on both the temporal and spatial dimensions. Wang et al. [37] proposed a Memory In Memory (MIM) network and a corresponding recurrent block. The MIM module uses the differential signal between the adjacent cyclic states to simulate the non-stationary and nearly stationary characteristics of the spatiotemporal dynamics of two cascaded, self-updated memory modules. The above method combines convolution with RNN, extracts the information in both time and space dimensions, and significantly improves the prediction tasks for many of the natural spatiotemporal processes. However, when the cell blocks of these networks are in the state of transition, the historical state information they rely on is only from the previous moment. This dependence is based on the assumption that the described spatiotemporal process is an approximate first-order Markov process, or is transformed into a first-order Markov process by some transformation. However, in fact, the natural spatiotemporal process is often non-stationary and highly nonlinear.

3. Preliminaries

3.1. Spatiotemporal Predictive Learning

Assuming that there is a spatiotemporal process X, an observation device takes snapshots of the spatiotemporal process X according to a fixed time interval (the interval can be considered as 1) and generates an observation sequence (X_{t-2}, X_{t-1}, X_t, X_{t+1}, X_{t+2} …). The observation snapshot X_t at each moment is a grid of data that satisfy the snapshot $X_t \in R^{M\times N\times C}$ (where M and N denote the width and length of the grid, and C denotes the dimension of the observed data at each grid point). The spatiotemporal prediction learning is to predict the most probable sequence (X_{t+1}, X_{t+2}, … , X_{t+Q-1}, X_{t+Q}) of observation snapshots at Q moments in the future, given an observation sequence (X_{t-P+1}, X_{t-P+2}, … , X_{t-1}, X_t) of a spatiotemporal process X, which satisfies the following equation:

$$\hat{X}_{t+1},\ldots,\hat{X}_{t+Q} = \underset{X_{t+1},\ldots,X_{t+Q}}{argmax}\ p\left(\hat{X}_{t+1},\ldots,\hat{X}_{t+Q}\middle|\hat{X}_{t-P+1},\ldots,\hat{X}_t\right), \quad (1)$$

The spatiotemporal prediction learning has a wide range of application scenarios, such as traffic flow prediction, behavior prediction, video prediction and radar echo prediction. Taking radar echo prediction as an example, the observation snapshot X_t is a grayscale image with the width of M and the length of N (the number of channels C is 1). The spatiotemporal prediction learning is to predict the content of the radar echo map in the next few hours, according to the current nearest radar echo map sequence, to predict the future precipitation at the grid level.

3.2. Stacked ConvLSTM Network

The main difference between the convolutional LSMT (ConvLSTM) and the standard LSTM is that the ConvLSTM replaces the matrix multiplication operation of standard LSTM with a convolution operation, so that the input information and state information can maintain the spatial structure in the LSTM unit. Simultaneous modeling of the spatiotemporal structure of information and state information avoids the loss caused by compressing the tensor information into the vector information in the standard LSTMs. In the ConvLSTM, whether it is hidden state information H_t, state information C_t, input information X_t, and various gates i_t, g_t, f_t, o_t, all maintain a 3-dimensional tensor structure ($\in R^{M\times N\times C}$, M and N are the width and length of the image, C is the number of input channels or feature channels, and the 3-dimensional tensor structure can be understood as in the two-dimensional space M × N, each point is a feature vector of dimension C). For a point in a two-dimensional space, the convolution operation captures the information in the neighborhood space of the point. The concatenating operation fuses the current input state and historical state information to construct a new memory state and hidden state. The spatial structure of this point is preserved throughout the process. The specific state transition follows the following equation:

$$
\begin{aligned}
g_t &= \tanh\Big(W_{xg} * X_t + W_{hg} * H_{t-1} + b_g\Big),\\
i_t &= \sigma\Big(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \bigodot C_{t-1} + b_i\Big),\\
f_t &= \sigma\Big(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \bigodot C_{t-1} + b_f\Big),\\
o_t &= \sigma\Big(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \bigodot C_{t-1} + b_o\Big),\\
C_t &= f_t \bigodot C_{t-1} + i_t \bigodot g_t,\\
H_t &= o_t \bigodot \tan\mathrm{h}(C_t),
\end{aligned}
\tag{2}
$$

where σ is the sigmoid function; $*$ denotes the convolution operator; and $\bigodot$ denotes the Hadamard product. The meaning of g_t is the state correction quantity generated according to the current input X_t and the historical hidden state H_{t-1}. The state after the transition C_t is the result of the joint action of the historical state C_{t-1} and the state correction quantity g_t. The final hidden state H_t is obtained by the state C_t transformation, and the output gate o_t controls how much of the internal state information can be observed.

Multiple ConvLSTMs can be stacked and temporally concatenated to form more complex structures. Such models have been applied to solve many real-world spatiotemporal prediction problems. The four-layer stacked structure is generally used when using the ConvLSTM for spatiotemporal prediction. At each timestep, the input information will be transformed by four ConvLSTM units, and the last ConvLSTM unit outputs the expected prediction. Here, the first two ConvLSTM units are generally understood as the Encoder part of the convolutional neural network, and the last two ConvLSTM units are understood as the Decoder part of the convolutional neural network. Each ConvLSTM unit has its memory, which independently stores historical state information, and each ConvLSTM unit only obtains historical state information from the ConvLSTM unit of the same layer at the previous moment. Intuitively, this method of obtaining historical state information is passive to a certain extent, and fails to give full play to the active role of current input information in controlling the transfer of historical state information. Figure 1 shows the stacked architecture of the ConvLSTM network used for spatiotemporal prediction.

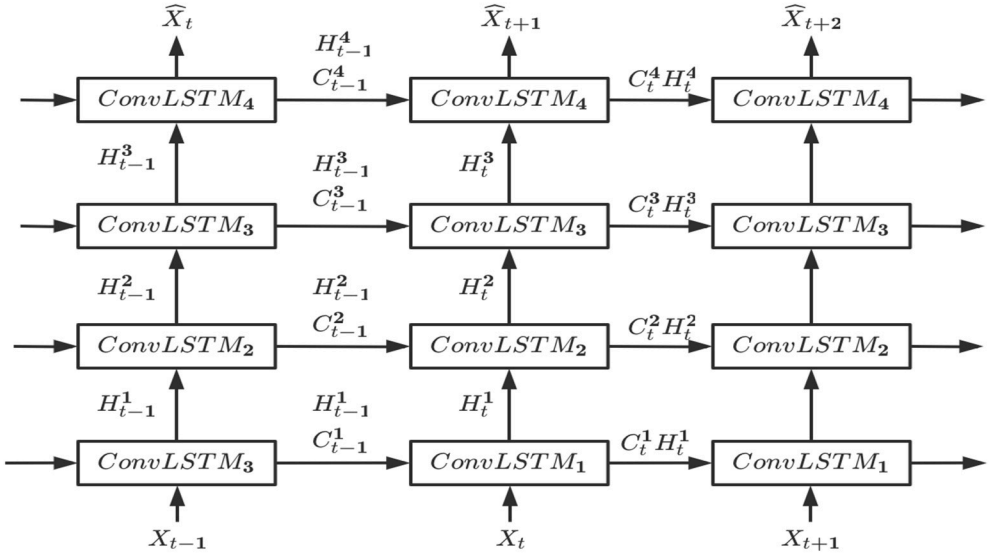

Figure 1. Stacked ConvLSTM recurrent neural network.

3.3. PredRNN

Compared with the ConvLSTM model [10], the biggest difference between the PredRNN model [35] is that the spatial memory unit is added to the PredRNN model. The proponents of the PredRNN model believe that each ConvLSTM unit of the ConvLSTM model only stores the memory state of the layer in the time dimension, and there is a lack of memory state transfer between the ConvLSTM units of each layer at the same time. This structure leads to a ConvLSTM model that can handle the dynamic information of spatiotemporal processes with time, but not the changes in the spatial details. In the ST-LSMT unit, a new spatial memory structure is added, combined with the original temporal memory structure in the ConvLSTM unit, to generate a new hidden state M^l_t. However, the spatial and temporal memory structures save and update their respective historical state information independently of each other. The newly added spatial memory structure mainly stores the spatial variations of the spatiotemporal process. It improves the memory flow by employing the zigzag path for better knowledge transfer, as the yellow line depicted in Figure 2. The architecture of the stacked recurrent neural network of the PredRNN model is shown in Figure 2.

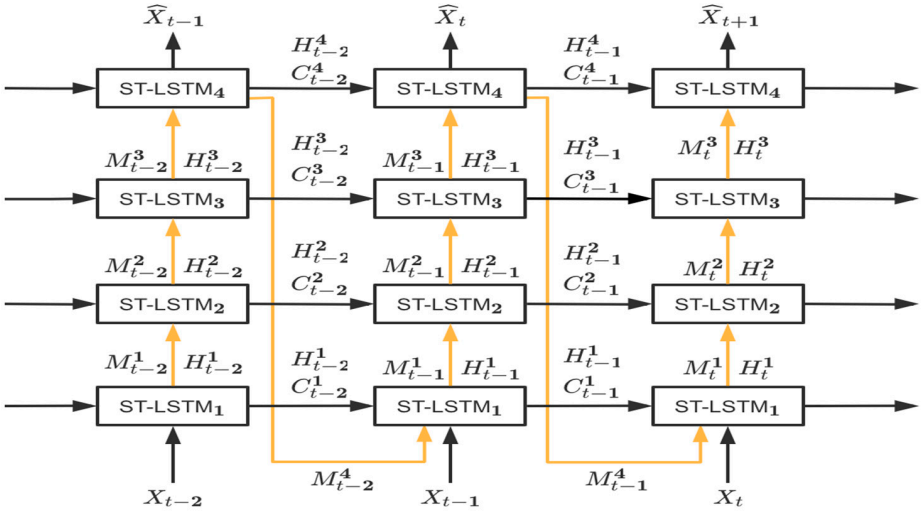

Figure 2. The stacked recurrent neural network of the PredRNN model (The yellow line is for spatial memory transfer, and the horizontal black line is for temporal memory transfer.).

Specifically, the state transition equations inside ST-LSTM are as follows:

$$
\begin{aligned}
g_t &= \tanh\left(W_{xg} * X_t + W_{hg} * H_{t-1}^l + b_g\right),\\
i_t &= \sigma\left(W_{xi} * X_t + W_{hi} * H_{t-1}^l + b_i\right),\\
f_t &= \sigma\left(W_{xf} * X_t + W_{hf} * H_{t-1}^l + b_f\right),\\
C_t^l &= f_t \bigodot C_{t-1}^l + i_t \bigodot g_t,\\
g_t' &= \tanh\left(W_{xg}' * X_t + W_{mg}' * M_{t-1}^l + b_g'\right),\\
i_t' &= \sigma\left(W_{xi}' * X_t + W_{mi}' * M_{t-1}^l + b_i'\right),\\
f_t' &= \sigma\left(W_{xf}' * X_t + W_{mf}' * M_{t-1}^l + b_f'\right),\\
M_t^l &= f_t' \bigodot M_{t-1}^l + i_t' \bigodot g_t',\\
o_t &= \sigma\left(W_{xo} * X_t + W_{ho} * H_{t-1}^l + W_{co} * C_t^l + W_{mo} * M_t^l + b_o\right),\\
H_t &= o_t \tanh \bigodot \left(W_{1\times 1} * \left[C_t^l, M_t^l\right]\right),
\end{aligned}
\tag{3}
$$

where σ denotes the sigmoid function; $*$ denotes the convolution operator; $\odot$ denotes the Hadamard product; and [ ] denotes the concatenation operator. g, i, f, and o denote the memory's state variations, input gate, forget gate, and output gate, respectively. C_t^l denotes the temporal memory unit, M_t^l denotes the spatial memory unit. The final output gate is controlled by the current input X_t, the hidden state H_{t-1}^l, the updated time state C_t^l, and the updated space state M_t^l. The final hidden state H_t is the synthesis of the updated time state C_t^l and the updated space state M_t^l.

Intuitively, the PredRNN model transmits state information more fluently than ConvLSTM. However, whether in temporal or spatial memory, PredRNN passively receives the state information of the previous moment, and lacks a mechanism to select the historical state information based on the current input activity.

4. Methods

As mentioned above, in the previous stacked recurrent neural networks used for spatiotemporal prediction, the passivity of the historical state transition brought about by relying only on the latest moment's historical state information has not been fully recognized. To overcome this passivity, it is first necessary to ensure that the historical state information is obtained from multiple moments in the past, and the second issue is how to obtain the historical state information at different moments. Considering these issues, we propose a Gated Attention LSTM (GA-LSTM) block to actively obtain the historical state we need from the state information of multiple past moments' information. In this chapter, we will first introduce the GA-LSTM block in detail, how to realize the initiative to obtain the historical state information, and then use the GA-LSTM block in combination with the commonly used stacked structure to construct a different RNN from the previous one. Intuitively, the new architecture of the network is different from the previous RNN architecture, in that each GA-LSTM unit no longer only obtains the state information from the previous moment but obtains the historical state information from multiple historical moments. The GA-LSTM block we propose is an improvement of the ST-LSTM block based on the PredRNN model, but it is feasible to apply the GA (Gated Attention) part to any unit similar to ConvLSTM.

4.1. GA-LSTM Block

In real natural spatiotemporal process, dynamic changes in the time dimension are usually very nonlinear, and the historical state that best matches the current input is often not the state information of the previous moment. For example, if a mouse shifts one step to the right at time t−2, one step downward at time t−1, and one step to the right at time t, then the motion state at time t is obviously more related to the state at time t−2. The previous stacked spatiotemporal prediction model lacks such a direct state transfer mechanism across time, so it cannot directly transfer the state across time and can only transfer the long-term state indirectly and iteratively. However, this iterative transfer will perform the filter based on the input information at that time in each iteration, and this filtering will largely filter out the information related to the current input.

The GA-LSTM block is mainly inspired by the attention mechanism in natural language processing. It no longer only passively depends on the state of the previous moment, such as the ST-LSTM block and the MIM block, but constructs a mechanism that relies on the state information of multiple past moments and makes an automatic selection. As shown in Figures 3 and 4 below, the historical state transition part of the ST-LSTM block and the MIM block in the sequential memory (the part covered by the dotted box in the figure) only depends on the state information at time t−1. The improvement of the MIM block relative to the ST-LSTM block is that, based on the assumption of differential stability, the state information is analyzed by using the differential information at time t−1 to perform stable information (corresponding to MIM-S in the figure) and unstable information (corresponding to MIM-N in the figure) decomposition. Compared with the single-time dependency of the ST-LSTM block and MIM block, we designed an LSTM-Atten module to process the state information of multiple moments in the past. The overall GA-LSTM block is shown in Figure 5.

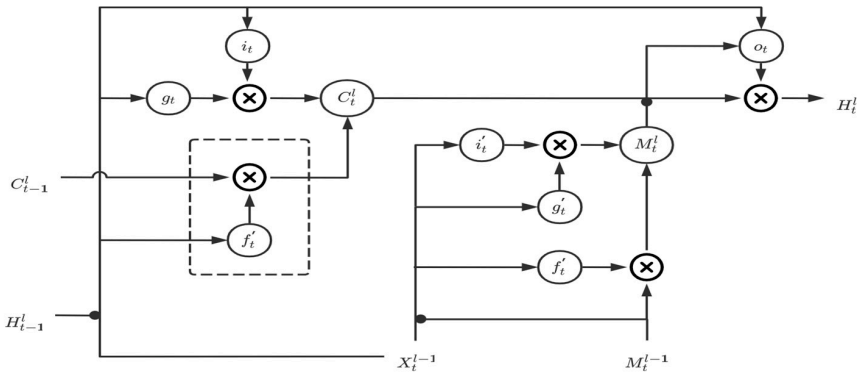

Figure 3. ST-LSTM block.

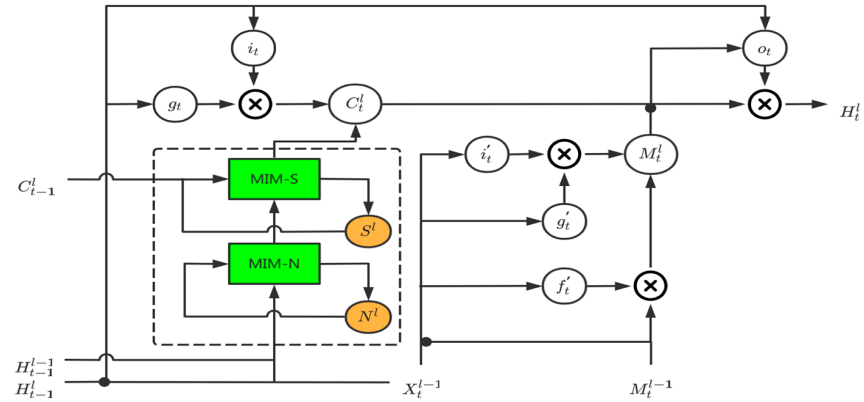

Figure 4. MIM block.

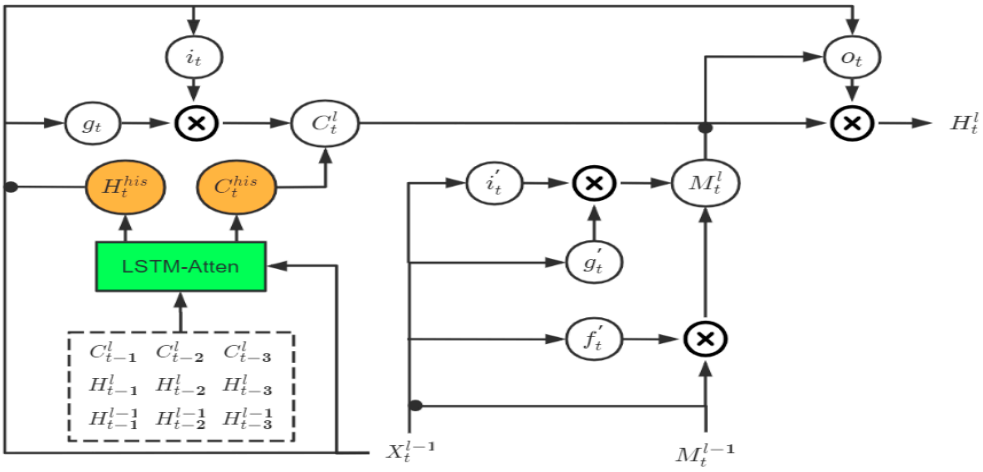

Figure 5. Proposed GA-LSTM block.

Here, the black dot denotes the concatenation operator. The input of the attention module LSTM-Atten includes the current input information X_t^{l-1} and the status information of the multiple past moments (3 moments as an example), including C_{t-1}^l, H_{t-1}^l, H_{t-1}^{l-1}, C_{t-2}^l, H_{t-2}^l, H_{t-2}^{l-1}, C_{t-3}^l, H_{t-3}^l, and H_{t-3}^{l-1}. Through the selection of the current input X_t^{l-1}, the outputs of the LSTM-Atten module are the expected historical state information C_t^{his}, and the expected historical hidden state information H_t^{his}. The overall transfer equation of GA-LSTM is written as:

$$C_t^{his},\ H_t^{his} = LSTM - Atten\left(X_t^{l-1}, C_{t-1}^l,\ H_{t-1}^l, H_{t-1}^{l-1} C_{t-2}^l, H_{t-2}^l,\ H_{t-2}^{l-1}, C_{t-3}^l,\ H_{t-3}^l, H_{t-3}^{l-1}\right),$$

$$g_t = \tanh\left(W_{xg} * X_t^{l-1} + W_{hg} * H_t^{his} + b_g\right),$$

$$i_t = \sigma\left(W_{xi} * X_t^{l-1} + W_{hi} * H_t^{his} + b_i\right),$$

$$f_t = \sigma\left(W_{xf} * X_t^{l-1} + W_{hf} * H_t^{his} + b_f\right),$$

$$C_t^l = C_t^{his} + i_t \bigodot g_t,$$

$$g_t' = \tanh\left(W_{xg}' * X_t^{l-1} + W_{mg}' * M_{t-1}^l + b_g'\right),$$

$$i_t' = \sigma\left(W_{xi}' * X_t^{l-1} + W_{mi}' * M_{t-1}^l + b_i'\right),$$

$$f_t' = \sigma\left(W_{xf}' * X_t^{l-1} + W_{mf}' * M_{t-1}^l + b_f'\right),$$

$$M_t^l = f_t' \bigodot M_{t-1}^l + i_t' \bigodot g_t',$$

$$o_t = \sigma\left(W_{xo} * X_t^{l-1} + W_{ho} * H_t^{his} + W_{co} * C_t^l + W_{mo} * M_t^l + b_o\right),$$

$$H_t = o_t \bigodot \tanh\left(W_{1\times1} * \left[C_t^l, M_t^l\right]\right), \quad (4)$$

where LSTM-Atten denotes the attention operation based on the LSTM structure. We think that the selection is attention. If the active variable X can selectively extract the passive variable, we can think that the active variable X exerts attention on the passive variable C. When the passive variable C has a greater positive effect on the active variable X to complete a specific task, the passive variable C will be given greater attention when it is extracted, and vice versa; it will be smaller or even negative. From this point of view, the LSTM structure is an attention mechanism. The active variable is the input information X_t^{l-1}, the passive variable is the state information H_{t-1}^l and C_{t-1}^l. The active variable X_t^{l-1}

selectively extracts the passive variable H_{t-1}^{l} and C_{t-1}^{l} to generate the H_t^l and C_t^l. Based on this understanding, we designed the LSTM-Atten module, using LSTM as the historical state information extraction operator integrated with the attention mechanism. The detailed structure of the LSTM-Atten module we designed is as follows (taking the extraction of historical state information from the past three moments as an example):

In Figure 6, the black dot denotes the concatenation operator. Combined with the selection mechanism of the LSTM structure, X_t^{l-1} is used as the active variable, the state variables H_{t-1}^{l} and C_{t-1}^{l} at time t−1 are actively selectively extracted to obtain H_t^{his1} and C_t^{his1}, the state variables H_{t-2}^{l} and C_{t-2}^{l} at time t−2 are actively selectively extracted to obtain H_t^{his2} and C_t^{his2}, and the state variables H_{t-3}^{l} and C_{t-3}^{l} at time t−3 are actively selectively extracted to obtain H_t^{his3} and C_t^{his3}. Then, use the convolutional network ENC_H to synthesize the hidden state information H_t^{his1}, H_t^{his2}, and H_t^{his3}, ENC_C to synthesize the state information C_t^{his1}, C_t^{his2}, and C_t^{his3}. Automatically assign the attention weights to the state information at each moment to generate the weighted state information H_t^{his} and C_t^{his} of gated attention. The internal state transition equation of the LSTM-Atten module is written as:

$$
\begin{aligned}
g_t^1 &= \tanh\left(W_{xg}^1 * X_t^{l-1} + W_{hg}^1 * H_{t-1}^l + W_{h1g}^1 * H_{t-1}^{l-1} + b_g^1\right),\\
i_t^1 &= \tanh\left(W_{xi}^1 * X_t^{l-1} + W_{hi}^1 * H_{t-1}^l + W_{h1i}^1 * H_{t-1}^{l-1} + b_i^1\right),\\
f_t^1 &= \tan h\left(W_{xf}^1 * X_t^{l-1} + W_{hf}^1 * H_{t-1}^l + W_{h1f}^1 * H_{t-1}^{l-1} + b_f^1\right)\\
o_t^1 &= \tan h\left(W_{xo}^1 * X_t^{l-1} + W_{ho}^1 * H_{t-1}^l + W_{h1o}^1 * H_{t-1}^{l-1} + b_o^1\right),\\
C_t^{his1} &= f_t^1 \bigodot C_{t-1}^l + i_t^1 \bigodot g_t^1,\\
H_t^{his1} &= o_t^1 \bigodot \tan h\left(W_{1\times1}^1 * C_t^{his1}\right),\\
g_t^2 &= \tanh\left(W_{xg}^2 * X_t^{l-1} + W_{hg}^2 * H_{t-2}^l + W_{h1g}^2 * H_{t-2}^{l-1} + b_g^2\right),\\
i_t^2 &= \tanh\left(W_{xi}^2 * X_t^{l-1} + W_{hi}^2 * H_{t-2}^l + W_{h1i}^2 * H_{t-2}^{l-1} + b_i^2\right),\\
f_t^2 &= \tan h\left(W_{xf}^2 * X_t^{l-1} + W_{hf}^2 * H_{t-2}^l + W_{h1f}^2 * H_{t-2}^{l-1} + b_f^2\right),\\
o_t^2 &= \tan h\left(W_{xo}^2 * X_t^{l-1} + W_{ho}^2 * H_{t-2}^l + W_{h1o}^2 * H_{t-2}^{l-1} + b_o^2\right)\\
C_t^{his2} &= f_t^2 \bigodot C_{t-2}^l + i_t^2 \bigodot g_t^2,\\
H_t^{his2} &= o_t^2 \bigodot \tan h\left(W_{1\times1}^2 * C_t^{his2}\right),\\
g_t^3 &= \tanh\left(W_{xg}^3 * X_t^{l-1} + W_{hg}^3 * H_{t-3}^l + W_{h1g}^3 * H_{t-3}^{l-1} + b_g^3\right),\\
i_t^3 &= \tanh\left(W_{xi}^3 * X_t^{l-1} + W_{hi}^3 * H_{t-3}^l + W_{h1i}^3 * H_{t-3}^{l-1} + b_i^3\right),\\
f_t^3 &= \tan h\left(W_{xf}^3 * X_t^{l-1} + W_{hf}^3 * H_{t-3}^l + W_{h1f}^3 * H_{t-3}^{l-1} + b_f^3\right),\\
o_t^3 &= \tan h\left(W_{xo}^3 * X_t^{l-1} + W_{ho}^3 * H_{t-3}^l + W_{h1o}^3 * H_{t-3}^{l-1} + b_o^3\right),\\
C_t^{his3} &= f_t^3 \bigodot C_{t-3}^l + i_t^3 \bigodot g_t^3,\\
H_t^{his3} &= o_t^3 \bigodot \tan h\left(W_{1\times1}^3 * C_t^{his3}\right),\\
C_t^{his} &= ENC\_C\left(C_t^{his1}, C_t^{his2}, C_t^{his3}\right)\\
H_t^{his} &= ENC\_H\left(H_t^{his1}, H_t^{his2}, H_t^{his3}\right)
\end{aligned}
\tag{5}
$$

where *ENC_C* and *ENC_H* denote the simple three-layer convolutional networks, respectively. Replacing the LSTM-Atten in Equation (4) with Equation (5), the state transition equation of the complete GA-LSTM block can be obtained.

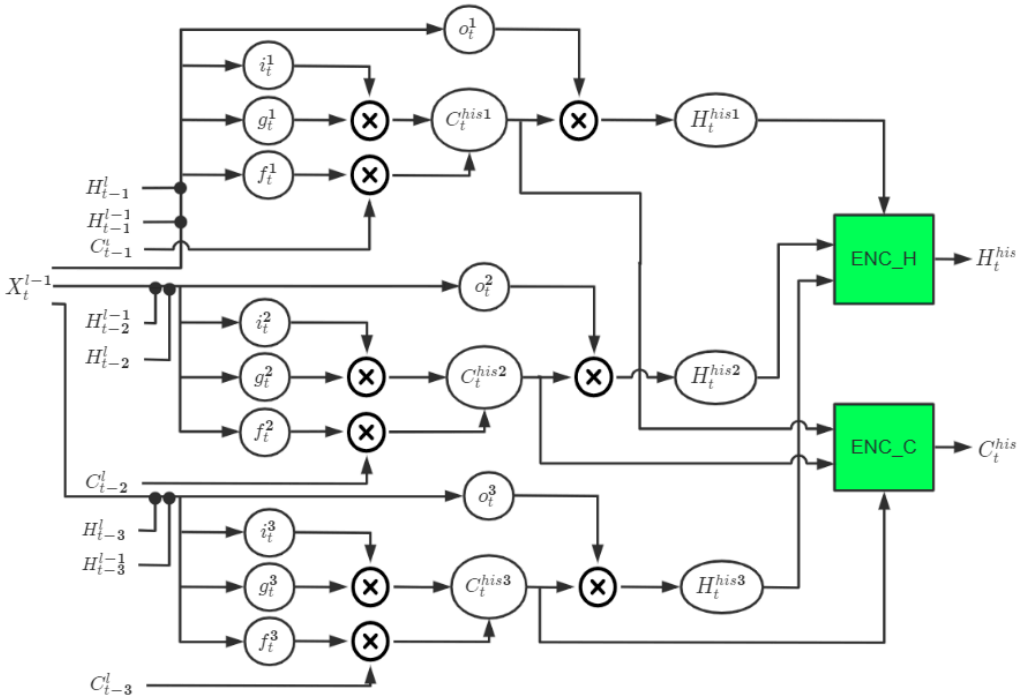

Figure 6. Proposed LSTM-Atten module.

Our proposed GA-LSTM is mainly reflected in two aspects: on the one hand, the LSTM structure is used to actively obtain the historical state information that matches the input information from multiple historical moments. On the other hand, two convolutional networks are used to automatically assign the attention weights to the historical state information matching the input, and to obtain the weighted gated attention state information, H_t^{his} and C_t^{his}.

4.2. Gated Attention Recurrent Neural Network

By stacking the GA-LSTM blocks, as shown in Figure 7, we propose the Gated Attention Recurrent Neural Network (GARNN) for spatiotemporal prediction. The most prominent feature of this recurrent neural network is that it no longer only depends on the state of the previous moment, but on the state of the past multiple moments.

The GARNN network inputs a frame at each timestep and outputs the predicted frame of the next moment. The yellow arrow is the state transition path of the spatial memory unit, and the black arrow is the state transition path of the temporal memory unit. Figure 7 shows the third layer of the GA-LSTM block at time t as an example. In addition to receiving the state information H_{t-1}^l and C_{t-1}^l at time $t-1$ (indicated by the black line), it also uses the state information H_{t-2}^l, C_{t-2}^l, H_{t-3}^l, and C_{t-3}^l at time $t-2$ and time $t-3$ (indicated by the blue line). In addition, inspired by the MIM model, the GARNN uses $W_1 * X_t^{l-1} + W_2 * H_{t-1}^{l-1}$ that can, to a certain extent, learn the transient information in the temporal dimension, and uses $W_3 * H_{t-1}^l + W_4 * H_{t-1}^{l-1}$ that can, to a certain extent, learn the transient information before and after the state transition. Adding this transient information helps to guide the extraction of the historical state information by input X, so the part of the red line in is added Figure 7.

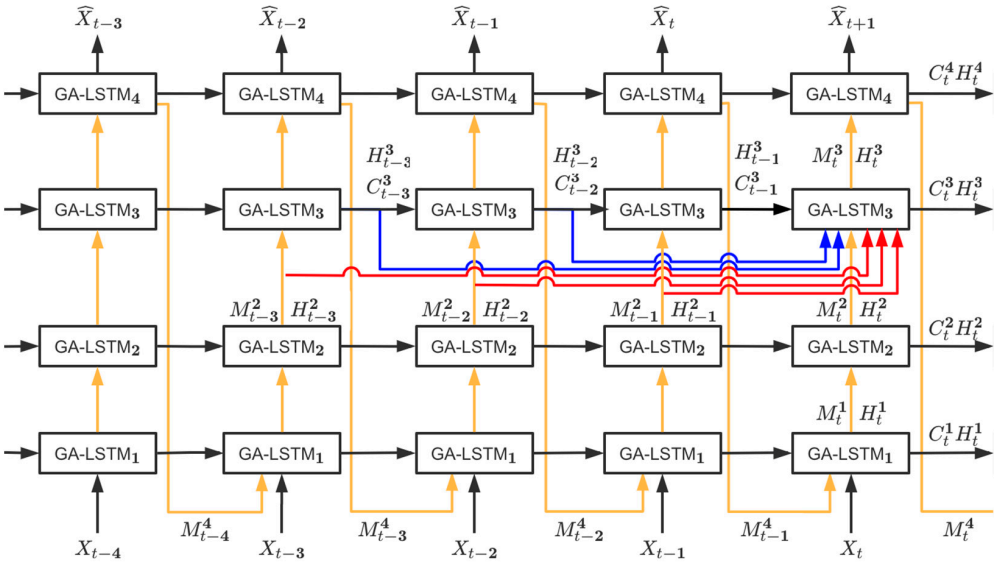

Figure 7. The main architecture of GARNN.

Furthermore, as shown in Figure 7, due to the use of a four-layer stacked structure, there will be four pieces of input information at each timestep (take time t as an example, the input information at this time includes X_t, H_t^1, H_t^2, and H_t^3), the last three pieces of input information are the hidden state information of the previous layer. Usually, the hidden state has a higher number of channels (such as 64 or 128 channels), and the number of channels of the first input is just the number of channels of the original image. To be consistent, we let a simple convolutional network transform the input information X before entering the GARNN network. Similarly, the output of the last layer of the GA-LSTM block is transformed by another simple convolutional network to the same number of channels as the original input.

5. Experiments

5.1. Experiment Design

We evaluated the proposed GARNN model for spatiotemporal prediction using a real-life weather radar echo dataset, which was given in the form of a 120 × 120 × 1 grayscale image (for the convenience of observation, when the radar echo image is displayed, it is converted into a color image using the color scale) with a coverage of 240 km × 240 km. The interval of each radar echo frame was 6 min. During training, every 20 time-consecutive frames were used as a set of samples, the first 10 frames were input (the period is exactly 1 h), and the last 10 frames were predicted (the period is also 1 h). During the inference, 10 frames in the next hour were predicted, based on the last 10 frames that could be obtained at the current moment.

As shown in Figure 8, we adopted a stacked recurrent neural network structure, which contains four-layer GA-LSTM units, and the number of feature channels of each gated structure in each GA-LSTM unit was 64. The model was trained with Mean Squared Error (MSE) as the loss function, using the Ranger optimizer, and the initial learning rate was set to 0.005. We set the mini-batch to eight, used the data-parallel mode for training, and set the time length to 3, extracting the historical state information of interest from the last 3 past moments each time. Additionally, we applied the layer normalization operation after the convolution of each gating operation of this model to reduce the covariate shift problem.

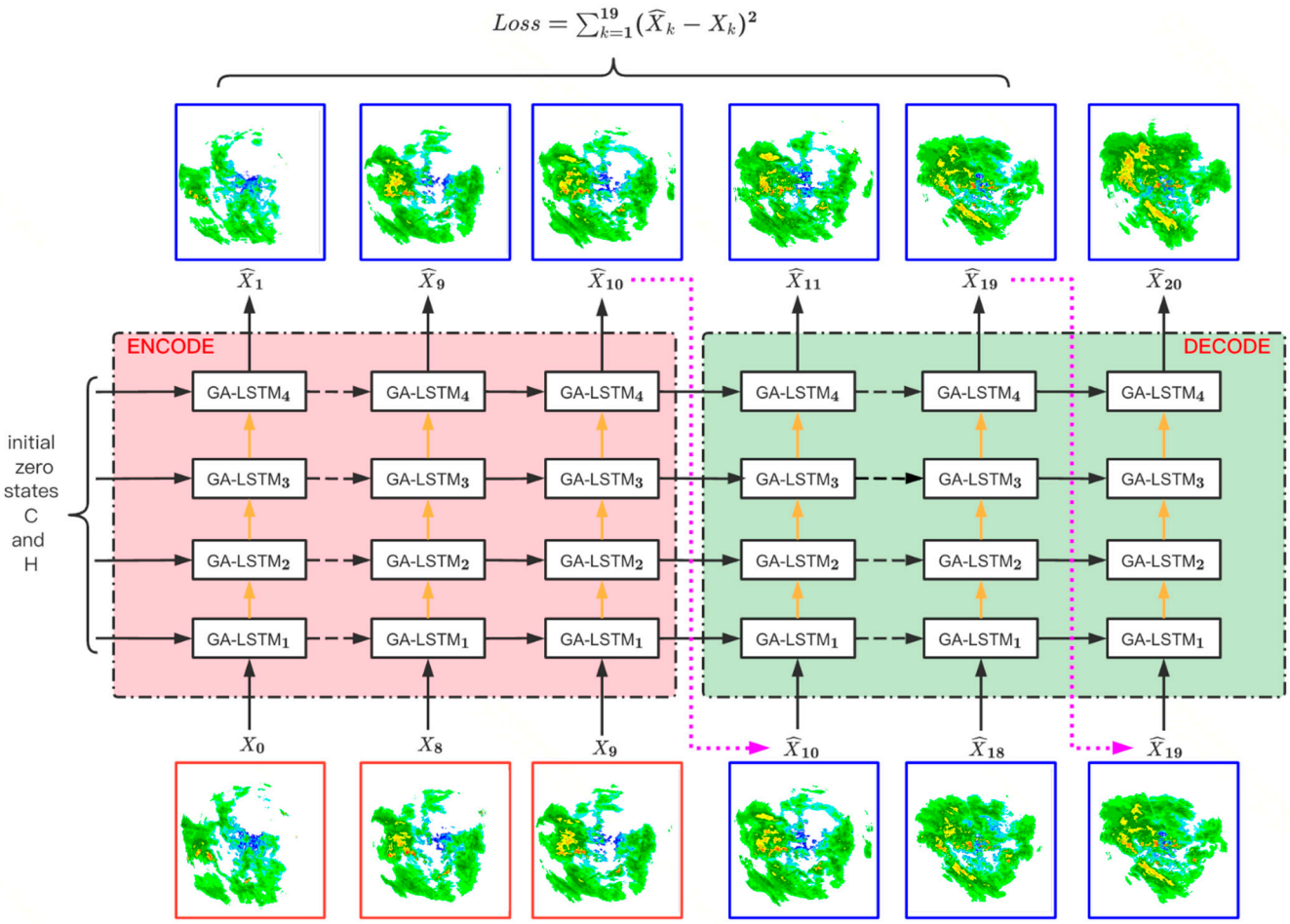

Figure 8. The flowchart of GARNN.

During the training and prediction, the image needs to be normalized to [0, 1], and then the mean of the squared differences of all the pixels was calculated as the MSE loss of the frame. The smaller the value, the better, indicating that the predicted image is closer to the ground truth image.

We usually pay more attention to the Critical Success Index (*CSI*) indicator in practical applications. After specifying the precipitation threshold, the CSI indicator reflects the accuracy of the prediction results in precipitation forecasting, and a higher *CSI* indicates a better precipitation forecasting accuracy. The *CSI* is defined as:

$$CSI = \frac{TP}{TP + FP + FN} \tag{6}$$

where *TP* corresponds to true positives; *FP* corresponds to false positives; and *FN* corresponds to false negatives; where positives can be considered as precipitation, and negatives can be considered as no precipitation.

5.2. Results

Figure 9 shows the comparison results of the GARNN model, the PredRNN model, and the ConvLSTM model on radar echo extrapolation prediction. As can be seen from the figure, relative to the ConvLSTM model, the PredRNN model and our proposed GARNN model predicted that the intensity attenuation of the generated echo was not apparent. The results of the PredRNN model show that it relied too much on spatial memory units, predicted that the echo changes in the generated images were small, and paid insufficient attention to the temporal memory units. The yellow echoes in the upper half of the Ground

Truth images show a variation pattern of "dissipating gradually from the middle and finally dissipating completely in the upper right half." The PredRNN model did not adequately capture this variation pattern, while the GARNN model captured it almost completely.

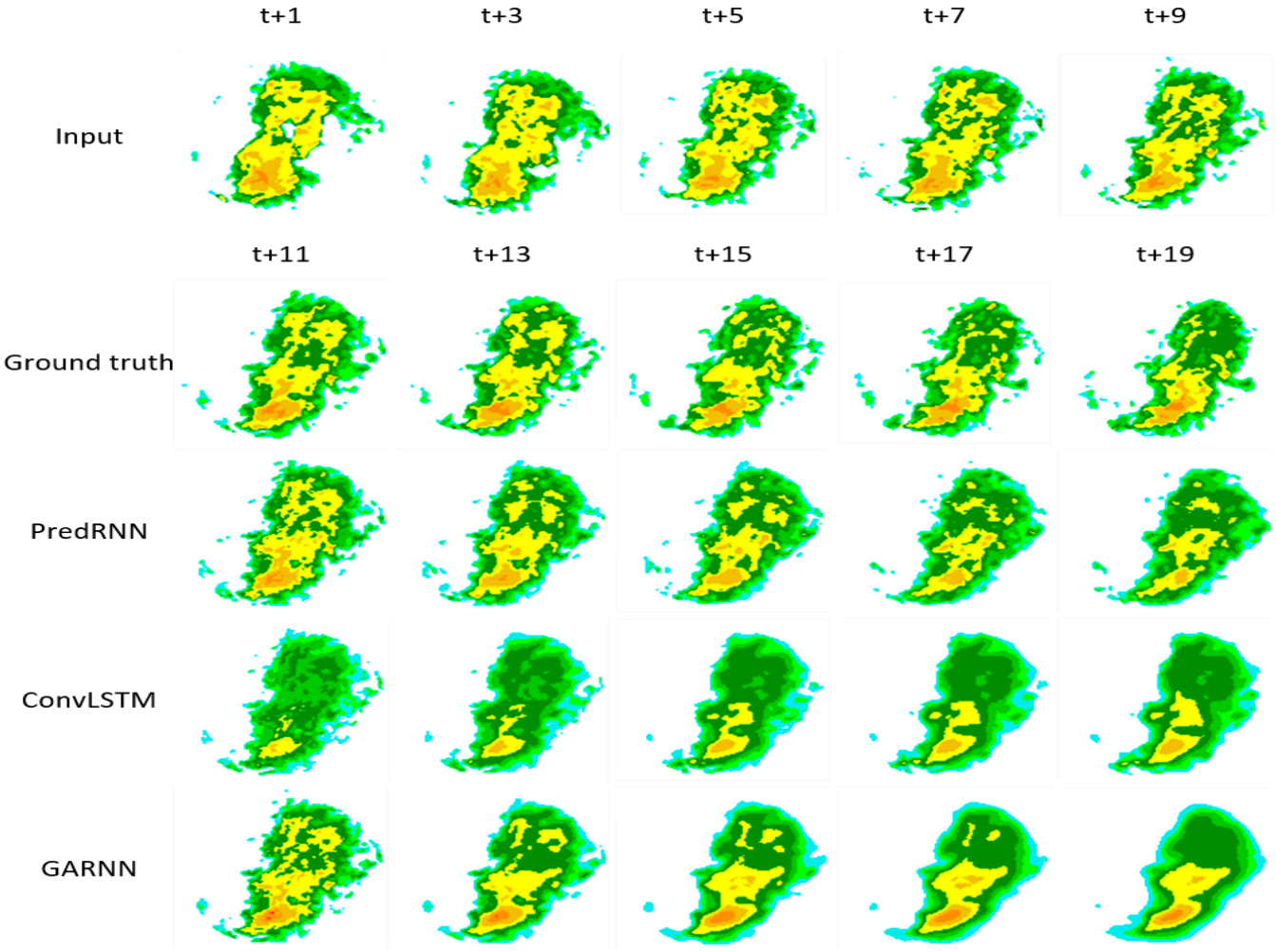

Figure 9. Example comparison of GARNN, ConvLSTM, and PredRNN predicting the echo map in the next 1 h.

As shown in Figure 10, the further the PredRNN model predicts, the greater the growth trend of MSE than the GARNN model. We speculate that the predictions of the PredRNN model are clearer than the GARNN model in the future, however, such clear details are not correct details, so the loss is too large, and the loss growth rate is hardly weakened. The GARNN model is relatively fuzzier, especially for the later predictions, and because it has learned the overall time dimension change characteristics, the overall MSE loss is smaller, and the loss growth rate is also lower. This can also be seen from the evaluation based on the CSI.

Combining the relationship between the radar and precipitation (according to our experience, the echo intensity threshold for determining precipitation is generally between [8,15] dBZ), we obtained three thresholds of 8 dBZ, 12 dBZ, and 15 dBZ for the CSI calculation. The overall mean value is shown in Table 1, and the CSI value comparisons are shown in Figures 11–13.

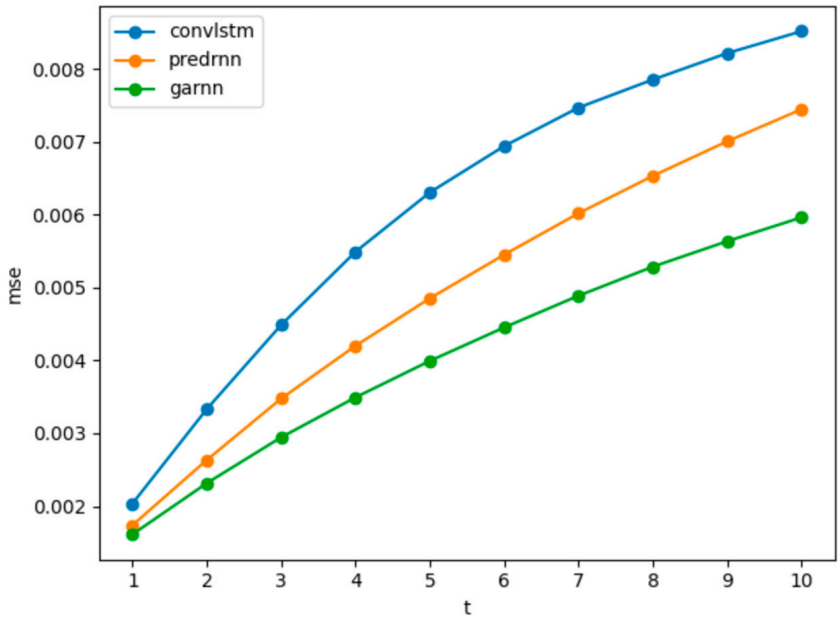

Figure 10. Frame-wise MSE of GARNN, ConvLSTM, and PredRNN.

Table 1. Quantitative results comparison of GARNN, ConvLSTM, and PredRNN.

Model	MSE	CSI-8	CSI-12	CSI-15
ConvLSTM	0.006063	0.612503	0.584735	0.5204
PredRNN	0.004934	0.637782	0.618641	0.591466
GARNN	0.004057	0.6604324	0.629499	0.595732

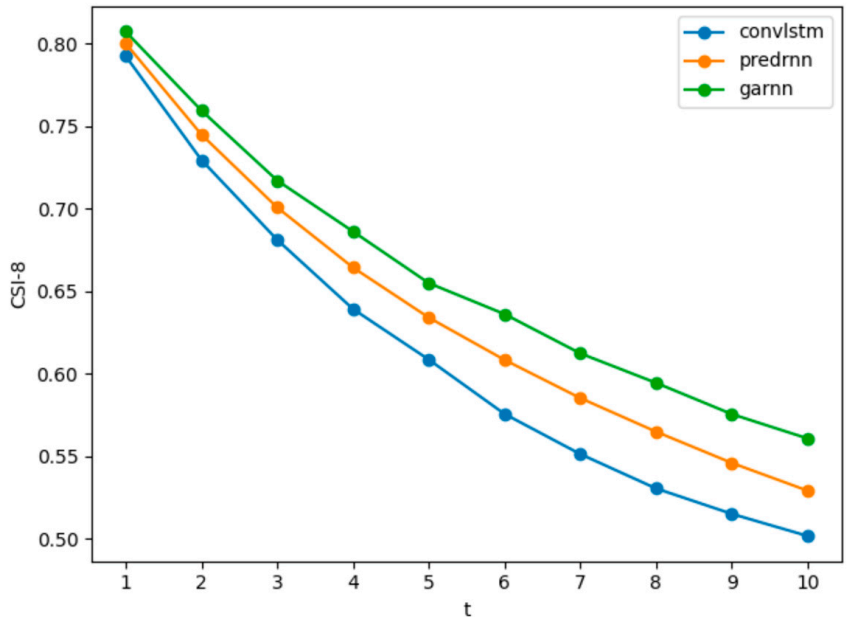

Figure 11. Frame-wise comparison of GARNN, ConvLSTM, and PredRNN at 8 dBZ threshold.

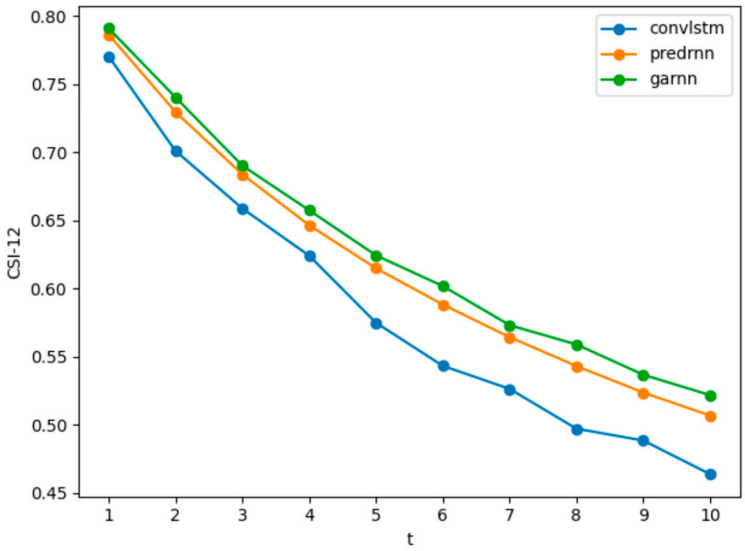

Figure 12. Frame-wise comparison of GARNN, ConvLSTM, and PredRNN at 12 dBZ threshold.

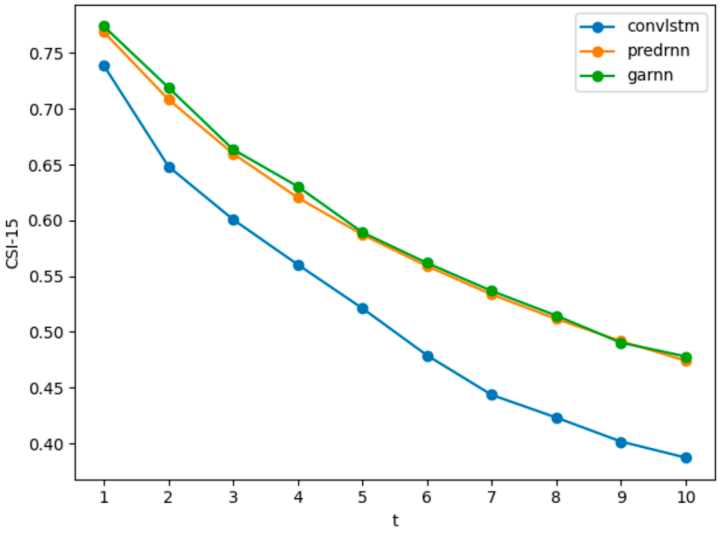

Figure 13. Frame-wise comparison of GARNN, ConvLSTM, and PredRNN at 15 dBZ threshold.

As can be seen in the figures above, at the three thresholds of 8 dBZ, 12 dBZ, and 15 dBZ, the GARNN model had the most noticeable improvement compared to the PredRNN model at 8 dBZ, while the two were almost the same at 15 dBZ.

This is mainly because the GARNN model reduces the dependence on the spatial memory unit, the detail retention ability is not as strong as that of the PredRNN model, and there is a certain strength attenuation, however, it learns better the dynamic variations in the time dimension, so its CSI values are better.

6. Conclusions

For spatiotemporal prediction problems, especially radar echo prediction, we have studied a series of methods to solve such problems and find that the spatiotemporal prediction models based on ConvLSTM work best. However, these methods currently use the first-order Markov properties to build a recurrent neural network by default, which is sufficient for simple motion, but lacks the capture of highly nonlinear and non-stationary motion features. This paper attempts to combine the attention mechanism and proposes a GA-LSTM unit, using the gated mechanism of the ConvLSTM unit. The GA-LSTM unit performs targeted screening of the multi-moment historical state information based on

the current input and then automatically assigns attention weights to them through a convolutional network. Using the GA-LSTM units, we built a recurrent neural network that satisfies the multi-order Markov properties and used this neural network to test a set of radar echo datasets. The test results showed that we learn better when it comes to dynamic variations in the time dimension, the prediction accuracy is greatly improved compared to the PredRNN model on which it is based. Nevertheless, there are still some problems that need to be further solved in our future research work. The computational complexity brought by the multi-moment historical information is relatively large, and the system resource overhead is too great.

Author Contributions: Conceptualization, G.W.; methodology, G.W. and W.C.; formal analysis, G.W. and W.C.; writing—original draft, G.W.; supervision, H.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MIST (Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program (IITP-2022-RS-2022-00156334) supervised by the IITP (Institute for Information and Communications Technology Planning and Evaluation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors also greatly appreciate the anonymous reviewers and academic editor for their careful comments and valuable suggestions to improve the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Prudden, R.; Adams, S.; Kangin, D.; Robinson, N.; Ravuri, S.; Mohamed, S.; Arribas, A. A review of radar-based nowcasting of precipitation and applicable machine learning techniques. *arXiv* **2020**, arXiv:2005.04988.
2. Cuomo, J.; Chandrasekar, V. Use of Deep Learning for Weather Radar Nowcasting. *J. Atmos. Ocean. Technol.* **2021**, *38*, 1641–1656. [CrossRef]
3. Ren, X.; Li, X.; Ren, K.; Song, J.; Xu, Z.; Deng, K.; Wang, X. Deep learning-based weather prediction: A survey. *Big Data Res.* **2021**, *23*, 100178. [CrossRef]
4. Liu, J.; Xu, L.; Chen, N. A spatiotemporal deep learning model ST-LSTM-SA for hourly rainfall forecasting using radar echo images. *J. Hydrol.* **2022**, *609*, 127748. [CrossRef]
5. Polson, N.G.; Sokolov, V.O. Deep learning for short-term traffic flow prediction. *Transp. Res. Part C Emerg. Technol.* **2017**, *79*, 1–17. [CrossRef]
6. Ali, A.; Zhu, Y.; Zakarya, M. Exploiting dynamic spatio-temporal correlations for citywide traffic flow prediction using attention based neural networks. *Inf. Sci.* **2021**, *577*, 852–870. [CrossRef]
7. Xiao, H.; Wang, C.; Li, Z.; Wang, R.; Bo, C.; Sotelo, M.A.; Xu, Y. UB-LSTM: A trajectory prediction method combined with vehicle behavior recognition. *J. Adv. Transp.* **2020**, *2020*, 8859689. [CrossRef]
8. Arashpour, M.; Ngo, T.; Li, H. Scene understanding in construction and buildings using image processing methods: A comprehensive review and a case study. *J. Build. Eng.* **2021**, *33*, 101672. [CrossRef]
9. Pradhyumna, P.; Shreya, G. Graph neural network (GNN) in image and video understanding using deep learning for computer vision applications. In Proceedings of the 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 4–6 August 2021; pp. 1183–1189.
10. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2015; p. 28.
11. Wang, Y.; Long, M.; Wang, J.; Gao, Z.; Yu, P.S. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Adv. Neural Inf. Processing Syst.* **2017**, *30*, 879–888.
12. Moskolaï, W.R.; Abdou, W.; Dipanda, A. Application of Deep Learning Architectures for Satellite Image Time Series Prediction: A Review. *Remote Sens.* **2021**, *13*, 4822. [CrossRef]
13. Hussein, E.A.; Ghaziasgar, M.; Thron, C.; Vaccari, M.; Jafta, Y. Rainfall Prediction Using Machine Learning Models: Literature Survey. *Artif. Intell. Data Sci. Theory Pract.* **2022**, *1006*, 75–108.
14. Mathieu, M.; Couprie, C.; LeCun, Y. Deep multi-scale video prediction beyond mean square error. In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May 2016.

15. Liu, W.; Luo, W.; Lian, D.; Gao, S. Future frame prediction for anomaly detection–a new baseline. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6536–6545.
16. Yi, Z.; Zhang, H.; Tan, P.; Gong, M. Dualgan: Unsupervised dual learning for image-to-image translation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2849–2857.
17. Liang, X.; Lee, L.; Dai, W.; Xing, E.P. Dual motion GAN for future-flow embedded video prediction. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1744–1752.
18. Kwon, Y.-H.; Park, M.-G. Predicting future frames using retrospective cycle gan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1811–1820.
19. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
20. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 3104–3112.
21. Xiang, Z.; Yan, J.; Demir, I. A rainfall-runoff model with LSTM-based sequence-to-sequence learning. *Water Resour. Res.* **2020**, *56*, e2019WR025326. [CrossRef]
22. Ranzato, M.; Szlam, A.; Bruna, J.; Mathieu, M.; Collobert, R.; Chopra, S. Video (language) modeling: A baseline for generative models of natural videos. *arXiv* **2014**, arXiv:1412.6604.
23. Srivastava, N.; Mansimov, E.; Salakhudinov, R. Unsupervised learning of video representations using lstms. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 843–852.
24. Babaeizadeh, M.; Finn, C.; Erhan, D.; Campbell, R.; Levine, S. Stochastic variational video prediction. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 30 April–3 May 2018.
25. Denton, E.; Fergus, R. Stochastic video generation with a learned prior. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1174–1183.
26. Lee, A.X.; Zhang, R.; Ebert, F.; Abbeel, P.; Finn, C.; Levine, S. Stochastic adversarial video prediction. *arXiv* **2018**, arXiv:1804.01523.
27. Wu, B.; Nair, S.; Martin-Martin, R.; Fei-Fei, L.; Finn, C. Greedy hierarchical variational autoencoders for large-scale video prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2318–2328.
28. Castrejon, L.; Ballas, N.; Courville, A. Improved conditional vrnns for video prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 7608–7617.
29. Franceschi, J.-Y.; Delasalles, E.; Chen, M.; Lamprier, S.; Gallinari, P. Stochastic latent residual video prediction. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 12–18 July 2020; pp. 3233–3246.
30. Villegas, R.; Pathak, A.; Kannan, H.; Erhan, D.; Le, Q.V.; Lee, H. High fidelity video prediction with large stochastic recurrent neural networks. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2019; p. 32.
31. Shi, X.; Gao, Z.; Lausen, L.; Wang, H.; Yeung, D.-Y.; Wong, W.-k.; Woo, W.-c. Deep learning for precipitation nowcasting: A benchmark and a new model. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5622–5632.
32. Chung, J.; Gulcehre, C.; Cho, K.H.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.
33. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* **2014**, arXiv:1409.1259.
34. Wang, Y.; Lu, J.; Ming, H.Y.; Li, J.L.; Long, M.; Fei-Fei, L. Eidetic 3D LSTM: A Model for Video Prediction and Beyond. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
35. Wang, Y.; Wu, H.; Zhang, J.; Gao, Z.; Wang, J.; Yu, P.; Long, M. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**. [CrossRef] [PubMed]
36. Wang, Y.; Gao, Z.; Long, M.; Wang, J.; Philip, S.Y. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5123–5132.
37. Wang, Y.; Zhang, J.; Zhu, H.; Long, M.; Wang, J.; Yu, P.S. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9154–9162.