atmosphere

MDPI

Article

Prediction of Short-Time Cloud Motion Using a Deep-Learning Model

Xinyue Su [1,*], Tiejian Li [1,2], Chenge An [1] and Guangqian Wang [1]

[1] State Key Laboratory of Hydroscience and Engineering, Tsinghua University, Beijing 100084, China; litiejian@tsinghua.edu.cn (T.L.); anchenge08@163.com (C.A.); dhhwgq@mail.tsinghua.edu.cn (G.W.)
[2] State Key Laboratory of Plateau Ecology and Agriculture, Qinghai University, Xining 810016, China
* Correspondence: suxy16@mails.tsinghua.edu.cn

Received: 24 August 2020; Accepted: 22 October 2020; Published: 26 October 2020

Abstract: A cloud image can provide significant information, such as precipitation and solar irradiation. Predicting short-time cloud motion from images is the primary means of making intra-hour irradiation forecasts for solar-energy production and is also important for precipitation forecasts. However, it is very challenging to predict cloud motion (especially nonlinear motion) accurately. Traditional methods of cloud-motion prediction are based on block matching and the linear extrapolation of cloud features; they largely ignore nonstationary processes, such as inversion and deformation, and the boundary conditions of the prediction region. In this paper, the prediction of cloud motion is regarded as a spatiotemporal sequence-forecasting problem, for which an end-to-end deep-learning model is established; both the input and output are spatiotemporal sequences. The model is based on gated recurrent unit (GRU)- recurrent convolutional network (RCN), a variant of the gated recurrent unit (GRU), which has convolutional structures to deal with spatiotemporal features. We further introduce surrounding context into the prediction task. We apply our proposed Multi-GRU-RCN model to FengYun-2G satellite infrared data and compare the results to those of the state-of-the-art method of cloud-motion prediction, the variational optical flow (VOF) method, and two well-known deep-learning models, namely, the convolutional long short-term memory (ConvLSTM) and GRU. The Multi-GRU-RCN model predicts intra-hour cloud motion better than the other methods, with the largest peak signal-to-noise ratio and structural similarity index. The results prove the applicability of the GRU-RCN method for solving the spatiotemporal data prediction problem and indicate the advantages of our model for further applications.

Keywords: cloud motion prediction; deep learning; gated recurrent unit; convolutional long short-term memory; satellite cloud image

1. Introduction

Recently, cloud-motion prediction has received significant attention because of its importance for the prediction of both precipitation and solar-energy availability [1]. Research has shown that the prediction of the short-time motion of clouds, especially of convective clouds, is important for precipitation forecasts [2–7]. Since most models of solar variability [8,9] and of solar irradiation [10–12] require cloud motion velocity as the main input, accurate cloud motion estimation is also essential for the intra-hour forecast of solar energy [13–16]. The difference between weather forecasts and solar forecasts is that the latter are usually conducted in a shorter time window (less than one hour). Otherwise, cloud-motion prediction is essentially similar in these two fields. Because the temperature of clouds is lower than that of the ground, clouds can be identified from infrared (IR) satellite images (with wavelengths of 10.5 to 12.5 μm) in which the intensity of IR radiation is correlated with

temperature [1,17]. Therefore, cloud motion can be estimated from a given sequence of IR images for weather forecasting [18] or intra-hour solar forecasting.

Nevertheless, cloud motion is a complex phenomenon involving nonrigid motion and nonlinear events [19], and predicting it remains challenging. Several methods have been proposed for the prediction of cloud motion; most of them are correspondence-based approaches. In general, cloud motion vectors (CMVs) are obtained by first locating salient image features, such as brightness gradients, corners, cloud edges, or brightness temperature gradients [20,21], and subsequently tracking these features in successive images with the assumption that they do not change significantly over a short interval. CMVs can be obtained from data collected by sky-imaging devices, such as whole-sky imagers (WSIs) [22], or by satellites. CMVs derived from WSI data are used for short-term forecasts (less than 20 min) in the local spatial area [11], whereas CMVs obtained from satellite images are commonly utilized to find the global atmospheric motion and the climate status of a large area [21,23]. Adopting a similar concept to CMVs, Brad and Letia [19] developed a model combining a block matching algorithm (BMA) and a best candidate block search, along with vector median regularization, to estimate cloud motion. This method divides successive images into blocks, restricting the candidate list of blocks to a predefined number, while in the full search BMA, the best match is found between the two blocks of successive frames in a full domain. Based on the idea of block matching, Jamaly and Kleissl [24] applied the cross-correlation method (CCM) and cross-spectral analysis (CSA) as matching criteria on cloud motion estimation. Additional quality-control measures, including removing conditions with low variability and less-correlated sites, can help to ensure that CSA and CCM reliably estimate cloud motion. Nevertheless, CCM can lead to relatively large errors because the assumption of uniform cloud motion does not hold in the presence of cloud deformation, topographically induced wind-speed variations, or a changing optical perspective [25]. This is a common problem for other block matching methods as well.

One approach to overcoming the challenges brought by variations in cloud motion is to compute the CMV of every pixel. Chow et al. [26] proposed a variational optical flow (VOF) technique to determine the subpixel accuracy of cloud motion for every pixel. They focused on cloud motion detection, and did not extend their work to prediction. Shakya and Kumar [27] applied a fractional-order optical-flow method to cloud-motion estimation and used extrapolations based on advection and anisotropic diffusion to make predictions. However, their method is not an end-to-end method of cloud-motion prediction.

Since the CMV is computed by extracting and tracking features, ameliorating feature extraction is another approach to improving performance. The deep convolutional neural network (CNN) [28] has proved able to extract and utilize image features effectively; it has achieved great success in visual recognition tasks, such as the ImageNet classification challenge [29]. Methods based on deep CNN have been introduced to cloud classification [30,31], cloud detection [32], and satellite video processing [33] in recent years. Although deep CNN has performed excellently when dealing with spatial data, it discards temporal information [34] that provides important clues in the forecasting of cloud motion. A prominent class of deep neural network called recurrent neural network (RNN) could learn complex and compound relationships in the time domain. However, the simple RNN model lacks the ability to backpropagate the error signal through a long-range temporal learning. Long short-term memory (LSTM) [35] was proposed to tackle this problem and this model is widely used in the solar power forecasting field [36,37]. Recent deep-learning models trained on videos have been used successfully for captioning and for encoding motion. Ji et al. [38] formulated a video as a set of images and directly applied deep CNN on the frames. Zha et al. [39] extended deep 2-D CNN to deep 3-D CNN and performed a convolutional operation on both the spatial and the temporal dimensions. Donahue et al. [40] combined convolutional networks with LSTM and proposed long-term recurrent convolutional network (LRCN). LRCN first processes the inputs with CNN and then feds the outputs of CNN into stacked LSTM. This method created a precedent on a combination of CNN and RNN regarded as recurrent convolutional network (RCN). Unlike previous proposals that

focused on high-level deep CNN "visual percepts", the novel convolutional long short-term memory (ConvLSTM) network proposed by Shi et al. [41] has convolutional structures in both the input-to-state and state-to-state transitions to extract "visual percepts" for precipitation now-casting. Ballas et al. [42] extended this work and proposed a variant form of the gated recurrent unit (GRU). They captured spatial information using an RNN with convolutional operation and empirically validated their GRU-RCN model on a video classification task. GRU-RCN has fewer parameters than ConvLSTM.

Since both the input and output of a cloud-motion forecast are spatiotemporal sequences, cloud-motion prediction is a spatiotemporal-sequence forecast problem for which GRU-RCN would seem well suited. However, Ballas et al. [42] focused on video classification, which is quite different from our forecast problem. Given the input video data, the output of their model is a number that depends on the class of the video; in our problem, the output should have a spatial domain as well. We need to modify the structure of the GRU-RCN model and apply it directly on the pixel level.

Moreover, there exists another challenge in the cloud motion prediction problem: new clouds often appear suddenly, at the boundary. To overcome this challenge, our model includes information about the surrounding context in which each small portion of the cloud is embedded; this was not considered in previous methods.

In this paper, we suggest the use of deep-learning methods to capture nonstationary information regarding cloud motion and deal with nonrigid processes. We propose a multiscale-input end-to-end model with a GRU-RCN layer. The model takes the surrounding context into account, achieves precise localization, and extracts information from multiple scales of resolution. Using a database of FenYun-2G IR satellite images, we compare our model's intra-hour predictions to those of the state-of-the-art variational optical-flow (VOF) method and three deep learning models (ConvLSTM, LSTM, and GRU); our model performs better than the other methods.

The remainder of this paper is organized as follows: Section 2 introduces the GRU-RCN model. Section 3 describes the data we used and the experiments we conducted. Section 4 presents the results, as well as briefly describes the other methods with which the GRU-RCN model was compared. Section 5 discusses the advantages and disadvantages of our model and our plans for future work. Section 6 provides our concluding remarks.

2. Methodology

2.1. Deep CNN

Deep CNNs [28] have been proven to extract and utilize image features effectively and have achieved great success in visual recognition tasks. Regular neural networks do not scale well to full images because, in the case of large images, the number of model parameters increases drastically, leading to low efficiency and rapid overfitting. The deep-CNN architecture avoids this drawback. The deep CNN contains a sequence of layers, typically a convolutional layer, a pooling layer, and a fully connected layer. In a deep CNN, the neurons in a given layer are not connected to all the neurons in the preceding layer but only to those in a kernel-size region of it. This architecture provides a certain amount of shift and distortion invariance.

2.2. GRU

A GRU [43] is a type of RNN. An RNN is implemented to process sequential data; it defines a recurrent hidden state, the activation of which depends on the previous state. Given a variable-length sequence $X = (x_1, x_2, \ldots x_t)$, the hidden state h_t of the RNN at each time step t is updated by:

$$h_t = \phi(h_{t-1}, x_t), \tag{1}$$

where ϕ is a nonlinear activation function.

An RNN can be trained to learn the probability distribution of sequences and thus to predict the next element in the sequence. At each time step t, the output can be represented as a distribution of probability.

However, because of the vanishing-gradient and exploding-gradient problems, training an RNN becomes difficult when input/output sequences span long intervals [44]. Variant RNNs with complex activation functions, such as LSTMs and GRUs, have been proposed to overcome this problem. LSTMs and GRUs both perform well on machine-translation and video-captioning tasks, but a GRU has a simpler structure and lower memory requirement [45].

A GRU compels each recurrent unit to capture the dependencies of different timescales adaptively. The GRU model is defined by the following equations:

$$z_t = \sigma(W_z x_t + U_z h_{t-1}), \tag{2}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1}), \tag{3}$$

$$\widetilde{h}_t = tanh(W x_t + U(r_t \odot h_{t-1})), \tag{4}$$

$$h_t = (1 - z_t) h_{t-1} + z_t \widetilde{h}_t, \tag{5}$$

where $\odot$ denotes element-wise multiplication; W, W_z, W_r and U, U_z, U_r are weight matrices; x_t is current input; h_{t-1} is the previous hidden state; z_t is an update gate; r_t is a reset gate; σ is the sigmoid function; $\widetilde{h}_t$ is a candidate activation, which is computed similarly to that of the traditional recurrent unit in an RNN; and h_t is the hidden state at time step t.

The update gate determines the extent to which the hidden state is updated when the unit updates its contents, and the reset gate determines whether the previous hidden state is preserved. More specifically, when the value of the reset gate of a unit is close to zero, the information from the previous hidden state is discarded and the update is based exclusively on the current input of the sequence. By such a mechanism, the model can effectively ignore irrelevant information for future states. When the value of the reset gate is close to one, on the other hand, the unit remembers long-term information.

2.3. GRU-RCN

In this section, we will introduce the GRU-RCN layer utilized in our model. A GRU converts input into a hidden state by fully connected units, but this can lead to an excessive number of parameters. In cloud imaging, the inputs of satellite images are 3-D tensors formed from the spatial dimensions and input channels. We regard the inputs as a sequence $X = (x_1, x_2, \ldots x_t)$; the size of the hidden state should be the same as that of the input. Let H, Wid, and C be the height, width, and number of the channels of input at every time step, respectively. If we apply GRU on inputs directly, the size of both the weight matrix W and the weight matrix U should be $H \times Wid \times C$.

Images are composed of patterns with strong local correlation that are repeated at different spatial locations. Moreover, satellite images vary smoothly over time: the position of a tracked cloud in successive images will be restricted to a local spatial neighborhood. Ballas et al. [42] embedded convolution operations into the GRU architecture and proposed the GRU-RCN model. In this way, recurrent units have sparse connectivity and can share their parameters across different input spatial locations. The structure of the GRU-RCN model is expressed in the following equations:

$$z_t^l = \sigma\left(W_z^l * x_t^l + U_z^l * h_{t-1}^l\right), \tag{6}$$

$$r_t^l = \sigma\left(W_r^l * x_t^l + U_r^l * h_{t-1}^l\right), \tag{7}$$

$$\widetilde{h}_t^l = \tan h(W^l * x_t^l + U^l * (r_t^l \odot h_{t-1}^l)), \tag{8}$$

$$h_t^l = \left(1 - z_t^l\right) h_{t-1}^l + z_t^l \widetilde{h}_t^l, \tag{9}$$

where $*$ denotes convolution and the superscript l denotes the layer of the GRU-RCN; the weight matrices W^l, W_z^l, W_r^l and U^l, U_z^l, U_r^l are 2-D convolutional kernels; and $h_t^l = h_t^l(i,j)$, where $h_t^l(i,j)$ is a feature vector defined at the location (i,j).

With convolution, the sizes of W^l, W_z^l, W_r^l and U^l, U_z^l, U_r^l are all $K_1 \times K_2 \times C$, where $K_1 \times K_2$ is the convolutional-kernel spatial size (chosen in this paper to be 3×3), significantly lower than that of the input frame $H \times Wid$. Furthermore, this method preserves spatial information, and we use zero padding in the convolution operation to ensure that the spatial size of the hidden state remains constant over time. The candidate hidden representation $\widetilde{h}_t^l(i,j)$, the activation gate $z_t^l(i,j)$, and the reset gate $r_t^l(i,j)$ are defined based on a local neighborhood of size $K_1 \times K_2$ at the location (i,j) in both the input data x_t^l and the previous hidden state h_{t-1}^l. In addition, the size of the receptive field associated with $h_t^l(i,j)$ increases with every previous timestep h_{t-1}^l, h_{t-2}^l ... as we go back in time. The model implemented in this paper is, therefore, capable of characterizing the spatiotemporal pattern of cloud motion with high spatial variation over time.

2.4. Multi-GRU-RCN Model

In this section, we will introduce the model structure of Multi-GRU-RCN. Ballas et al. [42] focused on the problem of video classification and therefore implemented a VGG16 model structure in their paper. However, this does not fit our problem well: we need to operate on the pixel level directly. The model structure of Shi et al. [41] consists of an encoding network as well as a forecasting network, and both networks are formed by stacking several ConvLSTM layers. In their model, there is a single input, and the input and output data have the same dimension. We modified this model structure and proposed a new one, which can extract information from the surrounding context. The model structure is presented in Figure 1.

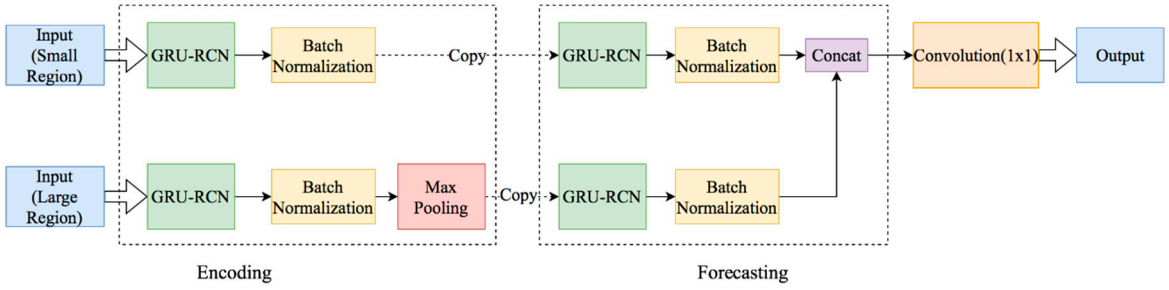

Figure 1. Outline of the multi-gated recurrent unit (GRU)-recurrent convolutional network (RCN) model.

There are multiple inputs in this model, and the input from each small region has the same center as the input from a larger region. The input from the small region has the same dimension as the output, while the input from the large region has four times as much area. We consider the region that is included in the large region but excluded in the small region as the surrounding context. The purpose of utilizing multiple inputs from different regions is to enrich the information with the surrounding context. Like the model of Shi et al. [41], our model consists of an encoding part and a forecasting part. In addition to stacked GRU-RCN layers, batch normalization [46] was introduced into both the encoding and forecasting parts to accelerate the training process and avoid overfitting. When utilizing input from a large region, we used a max pooling layer to reduce the dimension and improve the ability of the model to capture invariance information of the object in the image. The initial states of the forecasting part are copied from the final state of the encoding part. We concatenate the output states of the two inputs and subsequently feed this into a 1×1 convolutional layer with ReLU activation to obtain the final prediction results.

3. Experiment and Data

3.1. Experimental Setup

In this paper, we used a set of satellite data from 2018. For computational convenience, we first normalized the data to the range 0 to 1. We randomly selected data from 200 days as training data, data from 20 days as validation data, and data from another 20 days as test data. Because each image was too large (512 × 512 pixels) for training, we divided it into small patches, and set the patch sizes at 64 × 64 pixels and 128 × 128 pixels. Every 64 × 64 patch was paired with a 128 × 128 patch (the 64 × 64 patch being in the center region of the 128 × 128 patch). Thus, each 512 × 512 frame was divided into 16 pairs of patches. The patches were at the same location but across consecutive time steps. Because the average velocity of cloud motion in the training dataset is 14.35 m/s according to the FY-2G Atmospheric Motion Vector data, about 81.05% of the pixels in each patch can be tracked in the next time step's patch at the same location.

The data instances were partitioned into nonoverlapping sequences, each of which is n frames long. For each sequence, we used the first $n-1$ frames as input to predict the last frame. For example, the first sequence implements frames 1 to $n-1$ to predict frame n, and the second sequence implements frames $n+1$ to $2n-1$ to predict frame $2n$. To select the exact value of n, we designed a pretraining process. For each value of n between 2 and 10, we set the batch size to 32 and randomly selected 100 batches from the training dataset. We trained the model using these batches of data and computed the average mean squared error (MSE) among all batches and the running time. The results are presented in Figure 2.

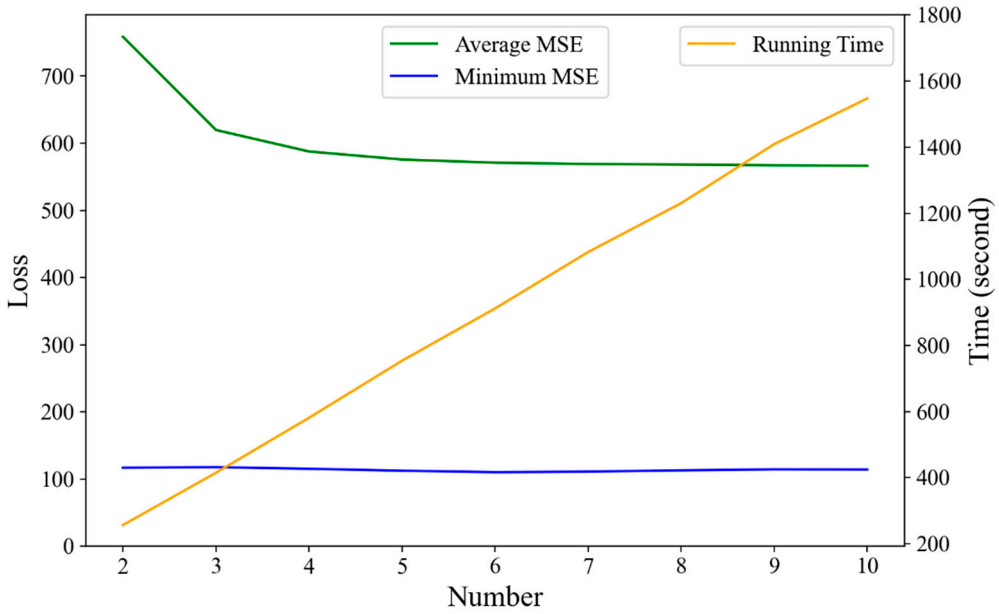

Figure 2. Results of average mean squared error (MSE), minimum MSE, and running time utilizing different values for the frame number n.

As seen in Figure 2, the running time increases almost linearly with the increase in n. The average MSE evidently falls as n increases from 2 to 6 but thereafter remains almost constant. The minimum MSE (i.e., the minimum batch MSE among all batches), however, is not very sensitive to the value of n. This indicates that when n is larger than 6, the running time increases as n increases, but this does not lead to a reduction in the MSE. Therefore, the value $n = 6$ was chosen for the experiments reported in this paper. The satellite collected data every hour; therefore, there were 12,800 cases in the training dataset, 1280 cases in the validation dataset, and 1280 cases in the test dataset.

The outline of the GRU-RCN layer is demonstrated in Figure 3. The output of the encoder will be implemented in the decoder for the production of the final output. We trained the GRU-RCN model by minimizing the MSE loss using backpropagation through time (BPTT) with a learning rate of 10^{-3}. The kernel size of the convolutional structure in our GRU-RCN layer was set to 3×3.

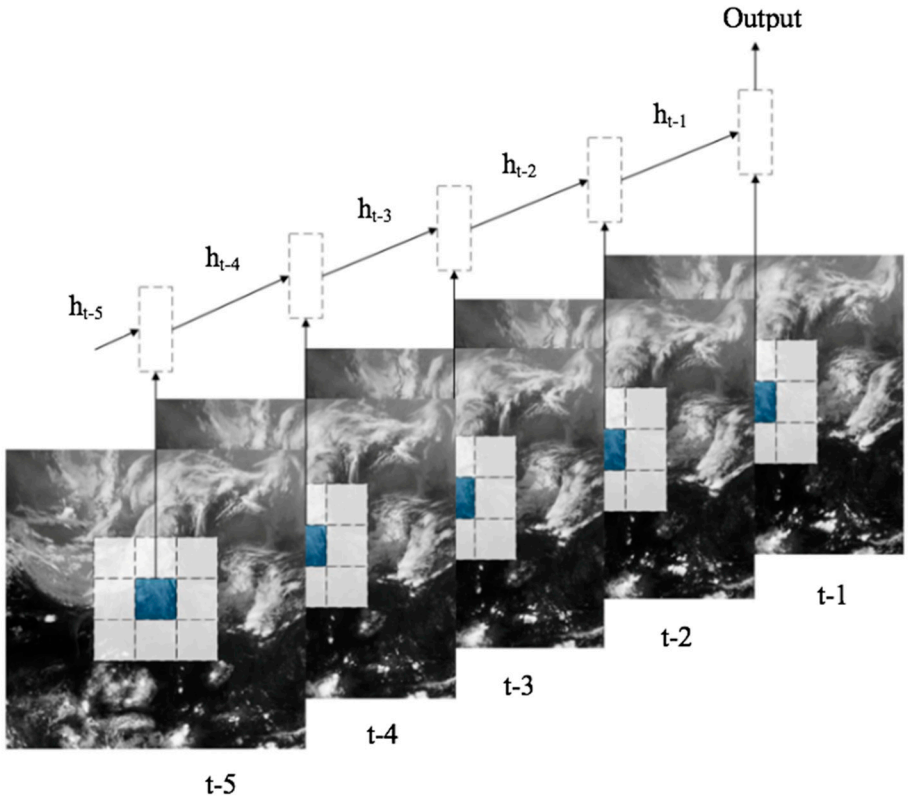

Figure 3. Outline of the gated recurrent unit (GRU)-recurrent convolutional network (RCN) layer applied to cloud-image prediction over time. Cloud images were obtained from the FY-2G IR1 database.

3.2. Test Benchmark

The performance of the proposed method was determined by two metrics: the peak signal-to-noise ratio (PSNR) [47] and structural similarity (SSIM) index [48]. PSNR is a widely used metric for evaluating the accuracy of algorithms. This metric indicates the reconstruction quality of the method used. The observed value at the prediction time step is not of practical relevance. Information regarding future events is not involved in the generation of the forecast satellite image; however, it still serves as a useful benchmark. The signal here was taken to be the observed value, whereas the noise was the error of the forecast image. The PSNR between the observed image f and predicted image g was defined as

$$PSNR(f,\ g)\ =\ 10\ log_{10}({I_{max}}^2/MSE(f,g)), \tag{10}$$

where $I_{max}\ =\ 2^8\ -\ 1$ is the maximum pixel intensity. $MSE(f,g)$ is the mean squared error between the observed and predicted image, defined as:

$$MSE(f,\ g)\ =\ \frac{1}{N}\sum_{i=1}^{N}(f_i - g_i)^2, \tag{11}$$

where N is the number of pixels in the satellite image.

For a smaller MSE, the PSNR will be larger, and, therefore, the algorithm accuracy will be higher.

SSIM is a quality assessment method used to measure the similarity between two images. It was proposed under the assumption that the quality perception of the human visual system (HVS) is correlated with structural information of the scene. Therefore, it considers image degradations as perceived changes in structural information, while PSNR estimates image degradations based on error sensitivity. The structural information is decomposed into three components: luminance, contrast, and structure. The SSIM between f and g is defined as:

$$SSIM(f,\ g)\ =\ l(f,\ g)c(f,\ g)s(f,\ g), \tag{12}$$

where $l(f,\ g)$, $c(f,\ g)$, and $s(f,\ g)$ are the luminance comparison, contrast comparison, and structure comparison between f and g, respectively:

$$l(f,\ g)\ =\ \frac{2\mu_f\mu_g + c_1}{{\mu_f}^2 + {\mu_g}^2 + c_1}, \tag{13}$$

$$c(f,\ g)\ =\ \frac{2\sigma_f\sigma_g + c_2}{{\sigma_f}^2 + {\sigma_g}^2 + c_1}, \tag{14}$$

$$s(f,\ g)\ =\ \frac{\sigma_{fg} + c_3}{\sigma_f\sigma_g + c_3}, \tag{15}$$

where μ_f and μ_g are the averages of f and g, σ_x are the variances of f and g, σ_{fg} is the covariance of f and g, and c_1, c_2, and c_3 are positive constants to stabilize the division with a zero denominator.

Besides, we also considered the mean bias error (MBE) as a supplementary metric. Although the value of MBE could not indicate the model reliability because the errors often compensate each other, it could show the degree to which the method underestimates or overestimates the results. With the purpose of exhibiting the degree more intuitively, the MBE was calculated as a percentage and the MBE between f and g was defined as

$$MBE(f,\ g)\ =\ \frac{1}{N}\sum_{i=1}^{N}\frac{g_i - f_i}{f_i} \times 100\%. \tag{16}$$

3.3. Data

FY-2, the first Chinese geostationary meteorological satellite, was launched on 31 December 2014, and positioned above the equator at 105° E. With the Stretched Visible and Infrared Spin Scan Radiometer (S-VISSR) on board, FY-2 can observe the Earth's atmosphere with high temporal and spatial resolutions. The IR1 channel (10.3~11.3 μm) China land-area images are obtained hourly for the spatial range 50° E~160° E, 4° N~65° N. The size of each image is 512 × 512 pixels, and the spatial resolution of each pixel is 13 km in both the north-south and east-west directions. The intensity of the pixels is 0~255. The relationship between the intensity count and brightness temperature is negative but not linear. An image instance of FY-2 is depicted in Figure 4.

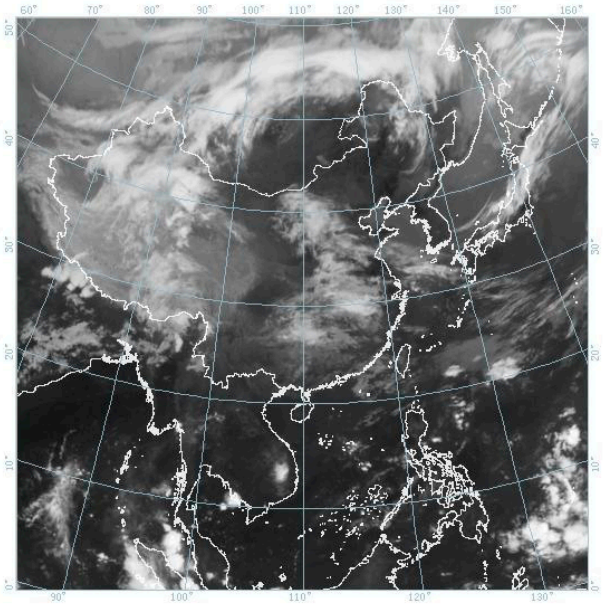

Figure 4. FY-2G IR1 China land-area image, produced at 12 a.m. on 1 March 2017.

4. Results and Analysis

One epoch is one training cycle through the entire training dataset. The models described in the previous sections were trained on the training dataset for 50 epochs and evaluated on the validation dataset after every epoch. The MSE loss is presented in Figure 5.

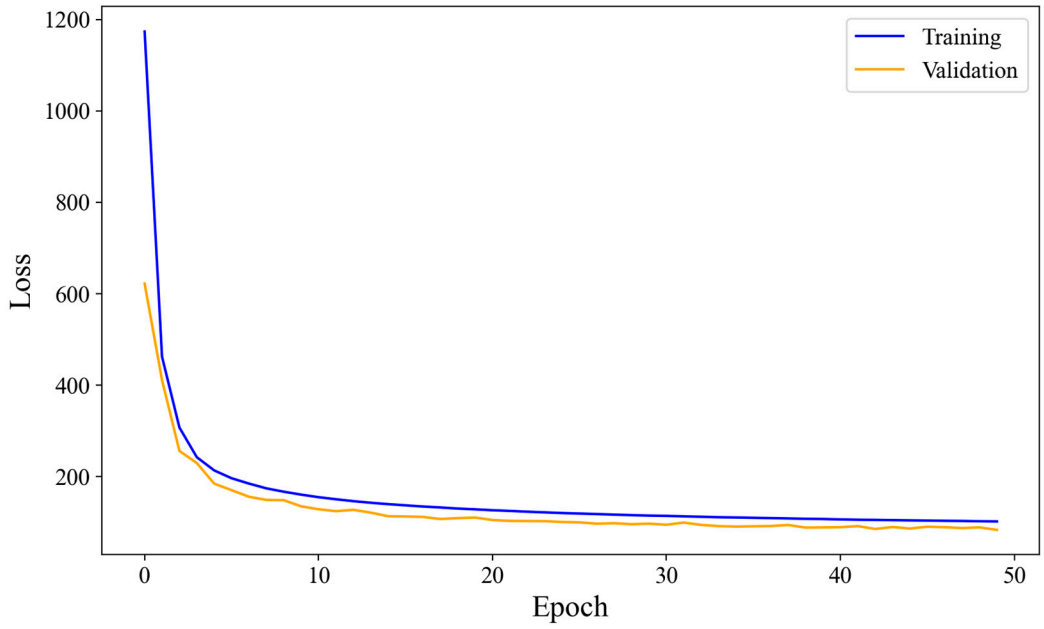

Figure 5. Mean squared error (MSE) loss in model training and validation.

In Figure 5, it is apparent that the MSE loss declined dramatically in the first 10 epochs; thereafter, the decline rate gradually decreased, and the MSE loss eventually converged to a lower level. When the training time was over 40 epochs, the loss was relatively small compared to that within the first 10. Despite the fluctuation of the validation loss, the integrated trend continued to decline, which indicates that the model was not overfitting. Thus, the training procedure was effective and converged to a quite satisfactory result.

We then randomly selected 20 days from 2018 as the test dataset, on which we compared our method (Multi-GRU-RCN) with the VOF technique, ConvLSTM, LSTM, and GRU. For the VOF algorithm, we used the method of Chow et al. [26], which minimizes the objective function by using brightness constancy and global smoothness as model assumptions to realize VOF. We set the size of the input patches as 128×128 and the size of the output patches as 64×64 to produce comparable results with Multi-GRU-RCN. For ConvLSTM, we adopted the model structure of Shi et al. [41], setting the kernel size at 3×3 for convolution. The input frame had the same size as the output frame. For LSTM and GRU, we deployed five frames to predict the next frame. Because LSTM and GRU cannot extract spatial information, we treated every pixel in a frame as an independent sample; thus, there were 4096 samples in a frame. All the experiments were carried out with NVIDIA Tesla T4 GPU. It takes 7.78 h to train the GRU model, 9.44 h to train the LSTM model, 12.29 h to train the ConvLSTM model, and 13.96 h to train the Multi-GRU-RCN model. There is no training process of the VOF method. In the test process, it requires 2.57, 3.65, 3.72, 4.28, and 4.73 s to predict one frame with the VOF method, GRU model, LSTM model, ConvLSTM model, and Multi-GRU-RCN model, respectively.

The MBEs predicted by VOF, GRU, LSTM, ConvLSTM, and Multi-GRU-RCN are 0.50%, 1.47%, 1.64%, −0.51%, and 0.45%. The nearly zero MBEs illustrate that none of these methods under or over forecast and no postprocessing steps are needed to calibrate the results. Quantitative results in terms of PSNR and SSIM over the test dataset are summarized in Table 1. The results shown in Table 1 confirm that Multi-GRU-RCN achieves the most promising results on both PSNR and SSIM metrics over the entire test dataset among these methods. To be specific, compared with ConvLSTM, Multi-GRU-RCN achieves a performance gain by 4.11% on PSNR and 2.60% on SSIM. In order to investigate the results in detail, we calculated the average PSNR and SSIM over the total 64 test samples for each day. The PSNR and the SSIM results using VOF, GRU, LSTM, ConvLSTM, and Multi-GRU-RCN on the test data for each day are compared in Figures 6 and 7, respectively.

Table 1. Comparison of Variation Optical-flow (VOF), Gated Recurrent Unit (GRU), Long Short-term Memory (LSTM), Convolutional Long Short-term Memory (ConvLSTM), and Multi-Gated Recurrent Unit (GRU)-Recurrent Convolutional Network (RCN) on the test dataset (highest measures are in bold).

Method	PSNR	SSIM
VOF	22.98	0.41
GRU	24.49	0.66
LSTM	24.58	0.66
ConvLSTM	28.45	0.77
Multi-GRU-RCN	**29.62**	**0.79**

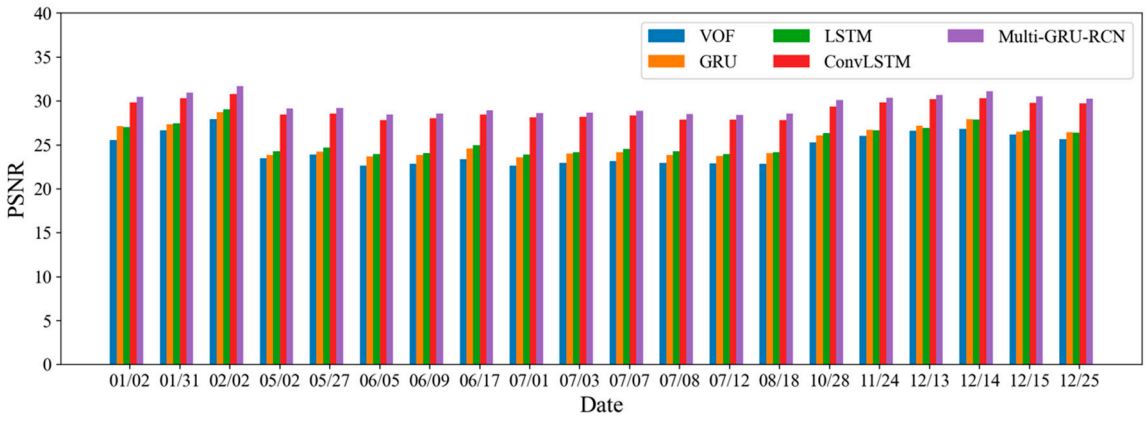

Figure 6. Peak Signal-to-noise Ratio (PSNR) when applying five different methods to the test data for various days.

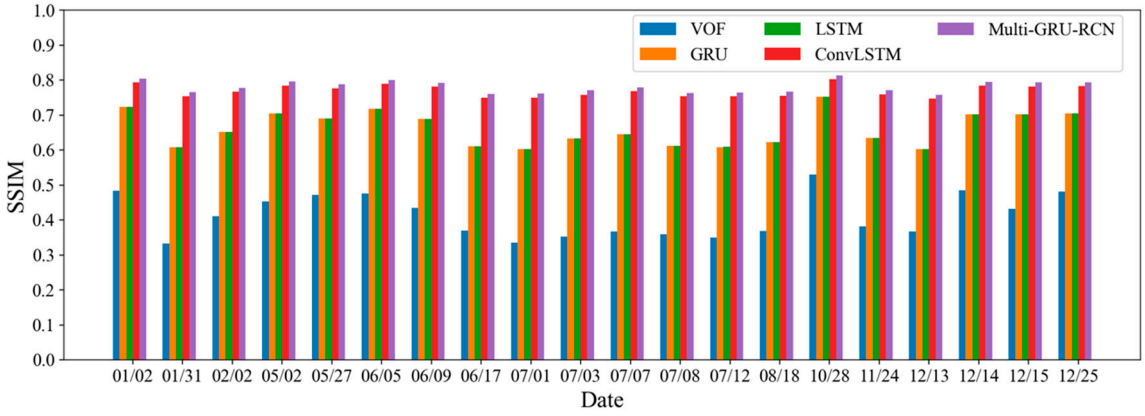

Figure 7. Structural Similarity (SSIM) when applying five different methods to the test data for various days.

According to Figures 6 and 7, the forecast results of these methods were consistent in terms of each metric. For instance, the PSNRs and SSIMs of the five methods were the highest on 2 February, which means that all four methods performed the best on the data of that day. Based on the forecast results, the Multi-GRU-RCN method consistently outperformed the other four methods during the whole computational time. The VOF was the worst-performing method on the test data. Multi-GRU-RCN and ConvLSTM had quite similar performance in terms of both MSE and PSNR values but Multi-GRU-RCN performed slightly better. The MSE of Multi-GRU-RCN forecasts on the test dataset was 72.93, which means that the average intensity difference per pixel between ground truth and prediction was 8.54 (a satisfactory result, given that the intensity range was 0~255).

To show the results more intuitively, we randomly picked three input sequences from the test dataset: May 2 between 0 and 5 am, January 31 between 6 and 11 pm, and July 7 between 6 and 11 am. Figure 8 shows the predictions of the next hour produced by VOF, GRU, LSTM, ConvLSTM, and Multi-GRU-RCN. The PSNRs predicted by VOF are 22.99, 23.01, and 22.91; those predicted by GRU are 23.92, 24.63, and 23.50; those predicted by LSTM are 23.98, 24.61, and 23.42; those predicted by ConvLSTM are 28.40, 28.50, and 27.67; and those predicted by Multi-GRU-RCN are 29.66, 30.37, and 29.86. The PSNR values predicted by Multi-GRU-RCN are consistently larger than those of the other methods, which indicates that its predictions are more accurate. This result also agrees with the difference between the ground truth and prediction. Even though the predictions by VOF have sharper outlines, the predictions by Multi-GRU-RCN have better accuracy. When a cloud appears at the edge of the prediction domain, Multi-GRU-RCN can predict it better than VOF. This proves that some of the complex spatiotemporal patterns in the dataset can be learned by the nonlinear and convolutional structure of the network. The model also performs well at predicting nonstationary processes, such as inversion and deformation, whereas VOF does not: in the VOF prediction for such situations, an abrupt change of intensity between adjacent pixels occurs at the bottom of the image. Multi-GRU-RCN gives a better prediction result without a blocky appearance.

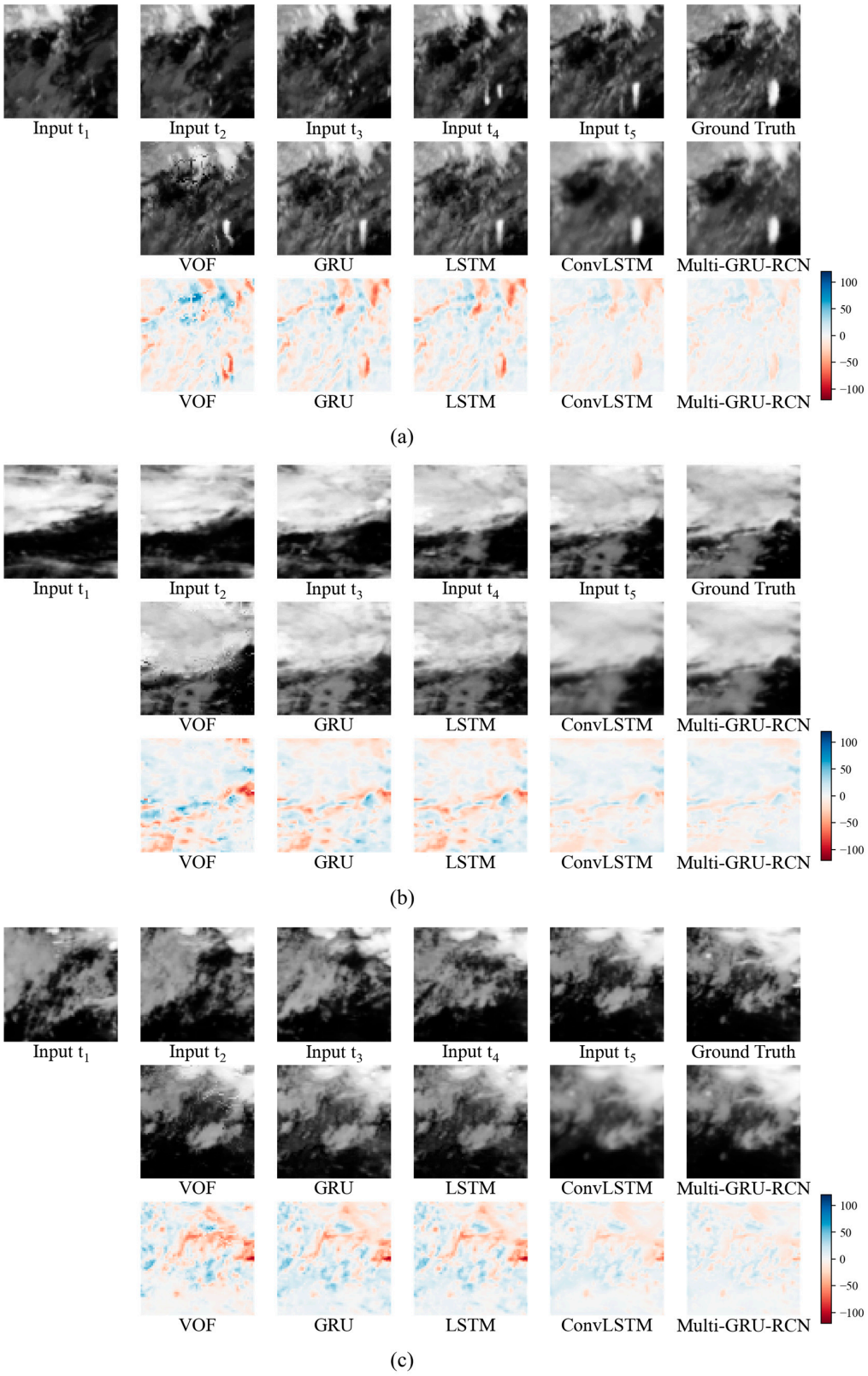

Figure 8. Three examples of satellite image and prediction results are shown in (**a**)–(**c**), respectively. In each panel, the first horizontal row shows the input sequence and the ground truth; the second horizontal row displays the predictions of five different methods; and the third horizontal row shows the difference between the prediction by each of the five methods and the ground truth.

5. Discussion

The relationship of GRU, LSTM, ConvLSTM, GRU-RCN, and Multi-GRU-RCN is illustrated in Figure 9. GRU is a simplification of LSTM by replacing the forget gate and input gate with the update gate, and combining the cell state and hidden state. Embedded convolutional operation in the recurrent unit, ConvLSTM, and GRU-RCN were implemented for spatial-temporal data and GRU-RCN has less parameters than ConvLSTM. As the GRU-RCN model structure proposed by Ballas et al. [42] focused on the video classification problem, we changed the model structure to adapt for the pixelwise cloud motion prediction problem. We considered the ConvLSTM model structure proposed by Shi et al. [41] as a reference, and replaced the ConvLSTM layer with the GRU-RCN layer. Besides, the surrounding context was introduced into our model to enrich input information.

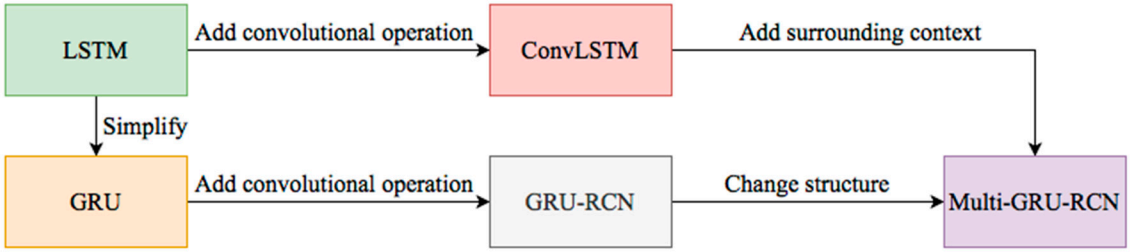

Figure 9. Relationship of Gated Recurrent Unit (GRU), Long Short-term Memory (LSTM), Convolutional Long Short-term Memory (ConvLSTM), Gated Recurrent Unit (GRU)-Recurrent Convolutional Network (RCN), and Multi-Gated Recurrent Unit (GRU)-Recurrent Convolutional Network (RCN).

In predicting cloud motion, both temporal and spatial information provide important clues. Temporally, the current frame correlates with the previous frame; spatially, the intensity of a given pixel correlates with those of the surrounding pixels. A GRU captures temporal information but ignores spatial information; therefore, it underperforms when compared to the ConvLSTM model, which captures both. However, in the ConvLSTM model, the input frame has the same shape as the output frame: as a result of the convolutional operation and same-padding method, it loses boundary information. In addition, the movement of the cloud is very complicated and cannot be determined by looking at the current region exclusively; more information must be brought into the model. To improve prediction accuracy, especially in the boundary region, we incorporated the surrounding context into our new end-to-end model. The performance improvement of Multi-GRU-RCN is also contributed to the model structure. For instance, in the experiment, we set the large region as the input and the small region as the output for the VOF algorithm, while we conducted a control experiment with the small region as both the input and output. The average PSNR and SSIM on the test dataset of the control experiment is 22.69 and 0.41, which indicates that the introduction of the large region only achieves a performance gain of 1.28% and 1.22% with the VOF algorithm. In the model structure aspect, we exploited the max pooling layer for dimension reduction and improved ability of the model to capture invariance information of the cloud while moving and fuse features from different scales. In addition, the activation functions introduce non-linearity into the model [49]. Accumulation of activation functions produces a promising model to learn sophisticated patterns. The essential advantage of the end-to-end structure is that all the parameters of the model can be simultaneously trained, making the training process more effective. The predictions of our model have consistently higher PSNR and SSIM than those of other methods. The spatial and temporal patterns learned by the model from the region of interest provides the fundamental of cloud motion. The utilization of external information out of the region of interest enriched the model understanding of environmental circumstances. This illustrates that utilizing information from both the internal and external region reveals a more accurate pattern of cloud motion.

There are three possible explanations for the better performance of Multi-GRU-RCN over the VOF algorithm. First, Multi-GRU-RCN can learn complex patterns during the training process. The clouds

often seem to appear instantaneously, indicating that they either derive from outside or are suddenly formed. If similar situations happened in the training dataset, Multi-GRU-RCN could learn these patterns during the training process and subsequently provide reasonable predictions in the test dataset. However, this could not be detected by the VOF algorithm. The second explanation is that Multi-GRU-RCN is trained end-to-end for this task. The VOF algorithm is not an end-to-end model, and it is difficult to find a reasonable way to update the future flow fields. The final reason is that Multi-GRU-RCN can smooth a blocky appearance, whereas the predictions of VOF will have a blocky appearance whenever there are abrupt changes in the motion vectors and therefore in the intensity between adjacent pixels.

Although the proposed Multi-GRU-RCN can achieve promising intra-hour cloud motion prediction, there are still limitations of this model. Compared with the VOF algorithm, the Multi-GRU-RCN produces blur prediction. This property is associated with the MSE loss when training the model. The future of the satellite cloud image is uncertain and by nature multimodal. When there are multiple valid outcomes with equally possibility, the MSE loss aims to accommodate the uncertainty by averaging all the possible outcomes, thus resulting in a blur prediction. Generative adversarial networks (GANs) have emerged as a powerful alternative to enhance prediction sharpness. In the future work, we will combine MSE loss with adversarial training loss to improve the visual quality of the predictions. Besides, limited by the number of layers in the architecture, the model could not totally eliminate the influence of interference, such as complex surface conditions. Li et al. [50] proposed a multi-scale convolutional feature fusion method for the cloud detection method. Their research confirmed that the usage of dilated convolutional layers and feature fusion of shallow appearance information and deep semantic information helps to improve the interference tolerance.

In this paper, the current forecasting range was an hour. The extension of the forecast time will convert the output from one frame to a sequence of frames. The weakness of the encoder-decoder architecture is that it lacks the alignment of the input and output sequence. Bahdanau et al. [51] proposed an attention mechanism utilizing a context vector to align the source and target inputs. The context vector preserves information from all hidden states in encoder cells and aligns them with the current target output. The attention mechanism allows the decoder to "attend" to different parts of the source sentence at each step of output generation; this concept has revolutionized the field. The introduction of the attention mechanism will address issues carried out in long-term horizon prediction. Furthermore, we plan to implement more data sources to enrich the information in the dataset and introduce data fusion techniques into the model improve the accuracy. Combining our current research with the precipitation forecast problem also merits further research.

6. Conclusions

In this paper, we introduced deep-learning methods into the field of cloud-motion prediction. This work is innovative, since traditional methods for cloud-motion prediction are mostly based on block matching and linear extrapolation, neglecting the nonstationary process during cloud movement. By formulating cloud-motion prediction as a spatial temporal data prediction problem, an end-to-end model with GRU-RCN was developed. Inclusion of the surrounding context enriched the information used. We tested this model's applicability on the cloud images of the FY-2G dataset for intra-hour prediction. Despite the relatively simple structure of our model, it can learn complex patterns through the nonlinear and convolutional structure of the network and works well when predicting the movement of clouds on a short timescale. This provides another example of the applicability of the GRU-RCN method in dealing with spatiotemporal data and learning complex patterns of images.

Author Contributions: Conceptualization, X.S. and T.L.; methodology, X.S. and T.L.; validation, X.S.; formal analysis, X.S.; writing—original draft preparation, X.S.; writing—review and editing, C.A.; visualization, X.S.; supervision, G.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China, grant numbers 2016YFE0201900 and 2017YFC0403600; the Xinjiang Production and Construction Corps, grant number 2017AA002; and the Sate Key Laboratory of Hydroscience and Engineering, grant number 2017-KY-04.

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

References

1. Mandal, A.; Pal, S.; De, A.; Mitra, S. Novel approach to identify good tracer clouds from a sequence of satellite images. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 813–818. [CrossRef]
2. Das, S.K.; Chanda, B.; Mukherjee, D.P. Prediction of cloud for weather now-casting application using Topology Adaptive Active Membrane. In *Proceedings of the 3rd International Conference on Pattern Recognition and Machine Intelligence, New Delhi, India*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 303–308.
3. Allen, M.R.; Ingram, W.J. Constraints on future changes in climate and the hydrologic cycle. *Nature* **2002**, *419*, 224–232. [CrossRef] [PubMed]
4. Held, I.M.; Soden, B.J. Robust Responses of the Hydrological Cycle to Global Warming. *J. Clim.* **2006**, *19*, 5686–5699. [CrossRef]
5. Mitchell, J.; Wilson, C.; Cunnington, W. On CO_2 climate sensitivity and model dependence of results. *Q. J. R. Meteorol. Soc.* **1987**, *113*, 293–322. [CrossRef]
6. Naegele, A.; Randall, D.A. Geographical and Seasonal Variability of Cloud-Radiative Feedbacks on Precipitation. *J. Geophys. Res. Atmos.* **2019**, *124*, 684–699. [CrossRef]
7. Muhammad, E.; Muhammad, W.; Ahmad, I.; Khan, N.M.; Chen, S. Satellite precipitation product: Applicability and accuracy evaluation in diverse region. *Sci. China Ser. E Technol. Sci.* **2020**, *63*, 819–828. [CrossRef]
8. Hoff, T.E.; Perez, R. Modeling PV fleet output variability. *Sol. Energy* **2012**, *86*, 2177–2189. [CrossRef]
9. Lave, M.; Kleissl, J. Cloud speed impact on solar variability scaling—Application to the wavelet variability model. *Sol. Energy* **2013**, *91*, 11–21. [CrossRef]
10. Chow, C.W.; Urquhart, B.; Lave, M.; Dominguez, A.; Kleissl, J.; E Shields, J.; Washom, B. Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed. *Sol. Energy* **2011**, *85*, 2881–2893. [CrossRef]
11. Marquez, R.; Gueorguiev, V.; Coimbra, C.F. Forecasting of Global Horizontal Irradiance Using Sky Cover Indices. *J. Sol. Energy Eng.* **2012**, *135*, 0110171–0110175.
12. Perez, R.; Kivalov, S.N.; Schlemmer, J.; Hemker, K.; Hoff, T.E. Short-term irradiance variability: Preliminary estimation of station pair correlation as a function of distance. *Sol. Energy* **2012**, *86*, 2170–2176. [CrossRef]
13. Bosch, J.; Kleissl, J. Cloud motion vectors from a network of ground sensors in a solar power plant. *Sol. Energy* **2013**, *95*, 13–20. [CrossRef]
14. Fung, V.; Bosch, J.L.; Roberts, S.W.; Kleissl, J. Cloud speed sensor. *Atmos. Meas. Tech. Discuss.* **2013**, *6*, 9037–9059. [CrossRef]
15. Huang, H.; Xu, J.; Peng, Z.; Yoo, S.; Yu, D.; Huang, D.; Qin, H. Cloud Motion Estimation for Short Term Solar Irradiation Prediction. In Proceedings of the IEEE International Conference on Smart Grid Communications, Vancouver, BC, Canada, 21–24 October 2013.
16. Quesada-Ruiz, S.; Chu, Y.; Tovar-Pescador, J.; Pedro, H.; Coimbra, C.F.M. Cloud-tracking methodology for intra-hour DNI forecasting. *Sol. Energy* **2014**, *102*, 267–275. [CrossRef]
17. Turiel, A.; Grazzini, J.; Yahia, H. Multiscale Techniques for the Detection of Precipitation Using Thermal IR Satellite Images. *IEEE Geosci. Remote Sens. Lett.* **2005**, *2*, 447–450. [CrossRef]
18. Vila, D.A.; Machado, L.A.T.; Laurent, H.; Velasco, I. Forecast and Tracking the Evolution of Cloud Clusters (ForTraCC) Using Satellite Infrared Imagery: Methodology and Validation. *Weather. Forecast.* **2008**, *23*, 233–245. [CrossRef]
19. Brad, R.; Letia, I.A. Cloud Motion Detection from Infrared Satellite Images. In Proceedings of the Second International Conference on Image and Graphics, Hefei, China, 31 July 2002; pp. 408–412.
20. Bedka, K.; Mecikalski, J.R. Application of Satellite-Derived Atmospheric Motion Vectors for Estimating Mesoscale Flows. *J. Appl. Meteorol.* **2005**, *44*, 1761–1772. [CrossRef]
21. Menzel, W.P. Cloud Tracking with Satellite Imagery: From the Pioneering Work of Ted Fujita to the Present. *Bull. Am. Meteorol. Soc.* **2001**, *82*, 33–47. [CrossRef]

22. Shields, J.E.; Karr, M.E.; Tooman, T.P.; Sowle, D.H.; Moore, S.T. The whole sky imager—A year of progress. In Proceedings of the Eighth Atmospheric Radiation Measurement (ARM) Science Team Meeting, Tucson, AZ, USA, 23–27 March 1998; pp. 677–685.
23. Hammer, A.; Heinemann, D.; Lorenz, E.; Lückehe, B. Short-term forecasting of solar radiation: A statistical approach using satellite data. *Sol. Energy* **1999**, *67*, 139–150. [CrossRef]
24. Jamaly, M.; Kleissl, J. Robust cloud motion estimation by spatio-temporal correlation analysis of irradiance data. *Sol. Energy* **2018**, *159*, 306–317. [CrossRef]
25. Yang, H.; Kurtz, B.; Nguyen, D.; Urquhart, B.; Chow, C.W.; Ghonima, M.; Kleissl, J. Solar irradiance forecasting using a ground-based sky imager developed at UC San Diego. *Sol. Energy* **2014**, *103*, 502–524. [CrossRef]
26. Chow, C.W.; Belongie, S.; Kleissl, J. Cloud motion and stability estimation for intra-hour solar forecasting. *Sol. Energy* **2015**, *115*, 645–655. [CrossRef]
27. Shakya, S.; Kumar, S. Characterising and predicting the movement of clouds using fractional-order optical flow. *IET Image Process.* **2019**, *13*, 1375–1381. [CrossRef]
28. Lecun, Y.; Bengio, Y. Convolutional Networks for Images, Speech, and Time Series. In *The Handbook of Brain Theory and Neural Networks*; MIT Press: Cambridge, MA, USA, 1995; Volume 3361, pp. 255–258.
29. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Neural Information Processing Systems*; ACM: Lake Tahoe, CA, USA, 2012.
30. Ye, L.; Cao, Z.; Xiao, Y.; Li, W. Ground-Based Cloud Image Categorization Using Deep Convolutional Visual Features. In Proceedings of the IEEE International Conference on Image Processing, Québec City, QC, Canada, 27–30 September 2015; pp. 4808–4812.
31. Ye, L.; Cao, Z.; Xiao, Y. DeepCloud: Ground-Based Cloud Image Categorization Using Deep Convolutional Features. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5729–5740. [CrossRef]
32. Shi, M.; Xie, F.; Zi, Y.; Yin, J. Cloud detection of remote sensing images by deep learning. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Beijing, China, 10–15 July 2016; pp. 701–704.
33. Xu, F.; Hu, C.; Li, J.; Plaza, A.; Datcu, M. Special focus on deep learning in remote sensing image processing. *Sci. China Inf. Sci.* **2020**, *63*, 140300. [CrossRef]
34. Wang, H.; Klaser, A.; Schmid, C.; Liu, C.-L. Action recognition by dense trajectories. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 20–25 June 2011.
35. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef] [PubMed]
36. Srivastava, S.; Lessmann, S. A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data. *Sol. Energy* **2018**, *162*, 232–247. [CrossRef]
37. Qing, X.; Niu, Y. Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. *Energy* **2018**, *148*, 461–468. [CrossRef]
38. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 221–231. [CrossRef]
39. Zha, S.; Luisier, F.; Andrews, W.; Srivastava, N.; Salakhutdinov, R. Exploiting Image-Trained CNN Architectures for Unconstrained Video Classification. *arXiv* **2015**, arXiv:1503.04144.
40. Donahue, J.; Hendricks, L.A.; Rohrbach, M.; Venugopalan, S.; Guadarrama, S.; Saenko, K.; Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 677–691. [CrossRef] [PubMed]
41. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.; Woo, W. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Proceedings of the Neural Information Processing Systems, Montréal, QC, Canada, 13 June 2015.
42. Ballas, N.; Yao, L.; Pal, C.; Courville, A. Delving Deeper into Convolutional Networks for Learning Video Representations. International Conference on Learning Representations. *arXiv* **2016**, arXiv:1511.06432.
43. Cho, K.; Van Merrienboer, 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.
44. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural. Comput.* **1997**, *9*, 1735–1780. [CrossRef]
45. Chung, J.; Gulcehre, C.; Cho, K.H.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.

46. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, arXiv:1502.03167.
47. Cuevas, E.; Zaldívar, D.; Perez-Cisneros, M.; Oliva, D. Block-matching algorithm based on differential evolution for motion estimation. *Eng. Appl. Artif. Intell.* **2013**, *26*, 488–498. [CrossRef]
48. Wang, Z.; Bovik, A.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans Image Process* **2004**, *13*, 600–612. [CrossRef]
49. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the International Conference on Machine Learning, Haifa, Israel, 21 June 2010.
50. Li, Z.; Shen, H.; Cheng, Q.; Liu, Y.; You, S.; He, Z. Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors. *ISPRS J. Photogramm. Remote Sens.* **2019**, *150*, 197–212. [CrossRef]
51. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.