remote sensing

MDPI

Article

ResTUnet: A Novel Neural Network Model for Nowcasting Using Radar Echo Sequences by Ground-Based Remote Sensing

Lei Zhang [1,2], Ruoyang Zhang [1,2], Yu Wu [3,*], Yadong Wang [1,2], Yanfeng Zhang [1,2], Lijuan Zheng [4], Chongbin Xu [5], Xin Zuo [5] and Zeyu Wang [3]

[1] Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475004, China; zhanglei@henu.edu.cn (L.Z.); tang@henu.edu.cn (Y.W.); zhangyanfeng@henu.edu.cn (Y.Z.)
[2] School of Computer and Information Engineering, Henan University, Kaifeng 475004, China
[3] Institute of Surface-Earth System Science, School of Earth System Science, Tianjin University, Tianjin 300072, China; 3021005058@tju.edu.cn
[4] Land Satellite Remote Sensing Application Center, Ministry of Natural Resources of China, Beijing 100048, China; zhenglijuan@lasac.cn
[5] Beijing Institute of Space Mechanics & Electricity, Beijing 100094, China; xucb_bisme@spacechina.com (C.X.); zuox1_bisme@spacechina.com (X.Z.)
* Correspondence: wu_yu@tju.edu.cn

Abstract: Radar echo extrapolation by ground-based remote sensing is essential for weather prediction and flight guiding. Existing radar echo extrapolation methods can hardly capture complex spatiotemporal features, resulting in the low accuracy of predictions, and, therefore, severely restrict their use in extreme weather situations. A deep learning method was recently applied for extrapolating radar echoes; however, its accuracy declines too quickly over a short time. In this study, we introduce a solution: Residual Transformer and Unet (ResTUnet), a novel model that improves prediction accuracy and exhibits good stability with a slow rate of accuracy decline. This presented Rest-Net model is designed to solve the issue of declining prediction accuracy by integrating a 1*1 convolution to diminish the neural network parameters. We constructed an observed dataset by Zhengzhou East Airport radar observation from July 2022 to August 2022 and performed 90 min experiments comprising five aspects, including extrapolation images, the Probability of Detection (POD) index, the Critical Success Index (CSI), the False Alarm Rate (FAR) index, and the Heidke Skill Score (HSS) index. The experimental results show that the ResTUnet model improved the CSI, HSS index, and the POD index by 17.20%, 11.97%, and 11.35%, compared to current models, including Convolutional Long Short-Term Memory (convLSTM), the Convolutional Gated Recurrent Unit (convGRU), the Trajectory Gated Recurrent Unit (TrajGRU), and the improved recurrent network for video predictive learning, the Predictive Recurrent Neural Network++ (predRNN++). In addition, the mean squared error of the ResTUnet model remains stable at 15% between 0 and 60 min and starts to increase after 60–90 min, which is 12% better than the current models. This enhancement in prediction accuracy has practical applications in meteorological services and decision making.

Keywords: radar echo; neural network model; weather prediction; remote sensing; climate change

check for updates

Citation: Zhang, L.; Zhang, R.; Wu, Y.; Wang, Y.; Zhang, Y.; Zheng, L.; Xu, C.; Zuo, X.; Wang, Z. ResTUnet: A Novel Neural Network Model for Nowcasting Using Radar Echo Sequences by Ground-Based Remote Sensing. *Remote Sens.* **2024**, *16*, 4792. https://doi.org/10.3390/rs16244792

Academic Editors: Sanghun Lim and Seongsim Yoon

Received: 11 November 2024
Revised: 17 December 2024
Accepted: 19 December 2024
Published: 23 December 2024

1. Introduction

Radar echo extrapolation of the atmosphere is primarily used for nowcasting. The World Meteorological Organization defined nowcasting as a description of current weather conditions and a weather forecast for the next 0–2 h in 1985 [1,2]. Nowcasting is mainly used to warn against hazardous weather events, which includes specifying the type, intensity, impact area, and timing of local hazardous weather, as well as detailed forecasts of precipitation, humidity, temperature, wind, clouds, visibility, and other routine information. There are two primary methods for radar extrapolation: physical equation-based radar extrapolation [3] and deep learning-based radar extrapolation [4]. Early radar echo extrapolation tracked the movement of echoes. Rinehart et al. [5] proposed using Tracking Radar Echo by Correlation (TREC) to track the motion of radar echoes, which can effectively invert the movement inside a storm. TREC technology has been widely used to estimate the motion field of echoes due to its good inversion ability. Li et al. [6] improved the TREC algorithm and proposed the Tracking Radar Echo by Correlation Based on Constraints and Variational Technique (COTREC), an effective objective analysis method used to smooth motion vectors and enforce continuity equations. COTREC corrects obvious errors that are typically caused by TREC failure, allowing us to identify regions of radar echo growth and decay. Mingxuan et al. [7] made improvements to TREC by calculating the optimal spatial cross-correlation and obtaining the convection storm motion vector of a two-dimensional array of equal size partitions of radar echoes or other data measured twice in a few minutes. Based on smoothing of the raw radar data, the pixels were subsequently grouped into blocks, and the optical flow method was used to calculate the velocity for each block [8]. Woo et al. [9] improved the optical flow method by proposing the variational optical flow method, which enhances the reflectivity within the selected range using a transformation function for feature tracking. These methods derived from the Lagrangian equation, which is deduced from certain physical assumptions and motion models. If these assumptions are inaccurate, the algorithm's results may not be accurate enough. In addition, this algorithm is highly sensitive to noise. When the noise is too large, the result will be seriously affected. To implement radar extrapolation algorithms, it is necessary to simplify and abstract the physical model to solve the Lagrangian equation [7]. However, the model's simplification and abstraction may result in information loss, affecting the accuracy of the algorithm. These drawbacks result in the low precision of results predicted by those algorithms.

To circumvent traditional method limitations, numerous scholars today are deploying deep learning into radar extrapolation. Deep learning exhibits an excellent capacity for nonlinear data fitting. Radar extrapolation methods based on deep learning are capable of learning through a vast volume of data and can better capture complex relationships in the data, which traditional techniques may not do. These methods can also effectively handle noise and interference in radar data and enhance extrapolation accuracy. Hence, numerous studies in recent years have employed deep learning in radar extrapolation [10]. Shi et al. [11] proposed the use of ConvLSTM for radar echo extrapolation. This method considers precipitation forecasting as a spatiotemporal sequence prediction problem. By extending Fully Connected Long Short-Term Memory (FC-LSTM) and incorporating convolutional structures inside and between internal states of the network model, this method can better capture spatiotemporal correlations. However, the convolutional recursive structure in the ConvLSTM model is position-invariant, while natural movements and transformations (such as rotation) often result in changes in position.

To address this issue, Shi et al. [12] proposed a modified TrajGRU model with a recursive structure that can actively learn the position variation structure in the cyclic connections, and the ability to capture high-level spatiotemporal features is higher than that of ConvLSTM. Wang et al. [13] introduced the PredRNN based on the RNN, which mainly consists of an Spatio-Temporal Long Short-Term Memory (ST-LSTM) unit that allows memory states belonging to different LSTMs to interact across layers. This greatly improves the accuracy of radar echo extrapolation. Afterwards, Wang et al. [14] proposed the PredRNN++, which introduces a gradient highway structure that provides an alternative, shorter path for the gradient flow to return from output to long-term input, enabling the PredRNN++ to dynamically capture short-term and long-term dependencies. Chen et al. [4] improved ConvLSTM by adding a Star-Shape Bridge structure for the transmission of features across time steps. Sønderby et al. [15] proposed a Neural Weather Model for Precipitation Forecasting (MetNet), which takes radar and satellite data as well as forecast lead time as input and generates probabilistic precipitation maps. The architecture uses axial self-attention to capture spatiotemporal features and can predict precipitation for the next 8 h at a high spatiotemporal resolution. U-Net was originally designed to tackle image segmentation problems and has since become a widely adopted architecture in the field of computer vision [16]. Han et al. [10] proposed a radar extrapolation method based on Unet [16], which first transforms the radar extrapolation problem into an image-to-image translation problem in deep learning. Their prediction results are comparable to the TrajGRU, and, due to the simplicity, efficiency, interpretability, and customizability of the UNet model, UNet-based models show great potential in solving time series applications [10]. Trebing et al. [17] proposed Small Attention-UNet Architecture (SmaAt-UNet), which is based on the well-known net architecture, is equipped with attention modules and depth-wise separable convolutions, and can predict relatively accurate results with fewer parameters. Espeholt et al. [18] proposed the MetNet-2 model, which can predict precipitation for the next 12 h at high resolution.

The input data of the ConvLSTM model goes through a convolutional layer at each time step, and the previous hidden state and memory state are used to update the current state at each time step [11]. This recursive update mechanism allows ConvLSTM to model time series data, but, due to the restriction of the convolutional kernel size, it can only capture local information of the data at each time step and is unable to capture global dependencies [12]. In the Unet-based model, skip connections are used between the encoder and decoder, allowing the decoder to access low-level features in the encoder [10]. This skip connection mechanism enables Unet to accurately reconstruct the input image. However, similarly limited by the convolutional kernel size, U-net can only capture local information of the data at each spatial position and is incapable of capturing global dependencies. Therefore, the ConvLSTM and U-net models are unable to capture global spatial-temporal features. This inability to recognize feature correlations beyond local temporal and spatial scales makes it challenging to effectively utilize the limited known information for longer-term extrapolation in the prediction process. As a result, this limitation leads to a rapid decline in the model's prediction accuracy over time.

To solve this problem, deep learning-based radar echo extrapolation methods outperform traditional approaches in terms of capturing temporal and spatial features and accuracy. Nevertheless, rapid prediction accuracy deterioration, slow prediction speed, and excessive model parameters hinder deep learning-based radar knowledge genera-tion. To address the mentioned challenges, we propose ResTUnet, a novel radar echo ex-trapolation model based on 1*1 convolutions and Residual Transformer (ResT) structures [19]. ResT structures facilitate enhanced processing of the encoding layer information in Unet, enhancing its ability to capture spatiotemporal features of the radar data sequence globally. We implement encoding and decoding layers in Unet through convolution and utilize 1*1 convolutions, replacing regular convolutions. This step is necessary to reduce the number of model parameters and improve training speed.

2. Model

2.1. Model Overview

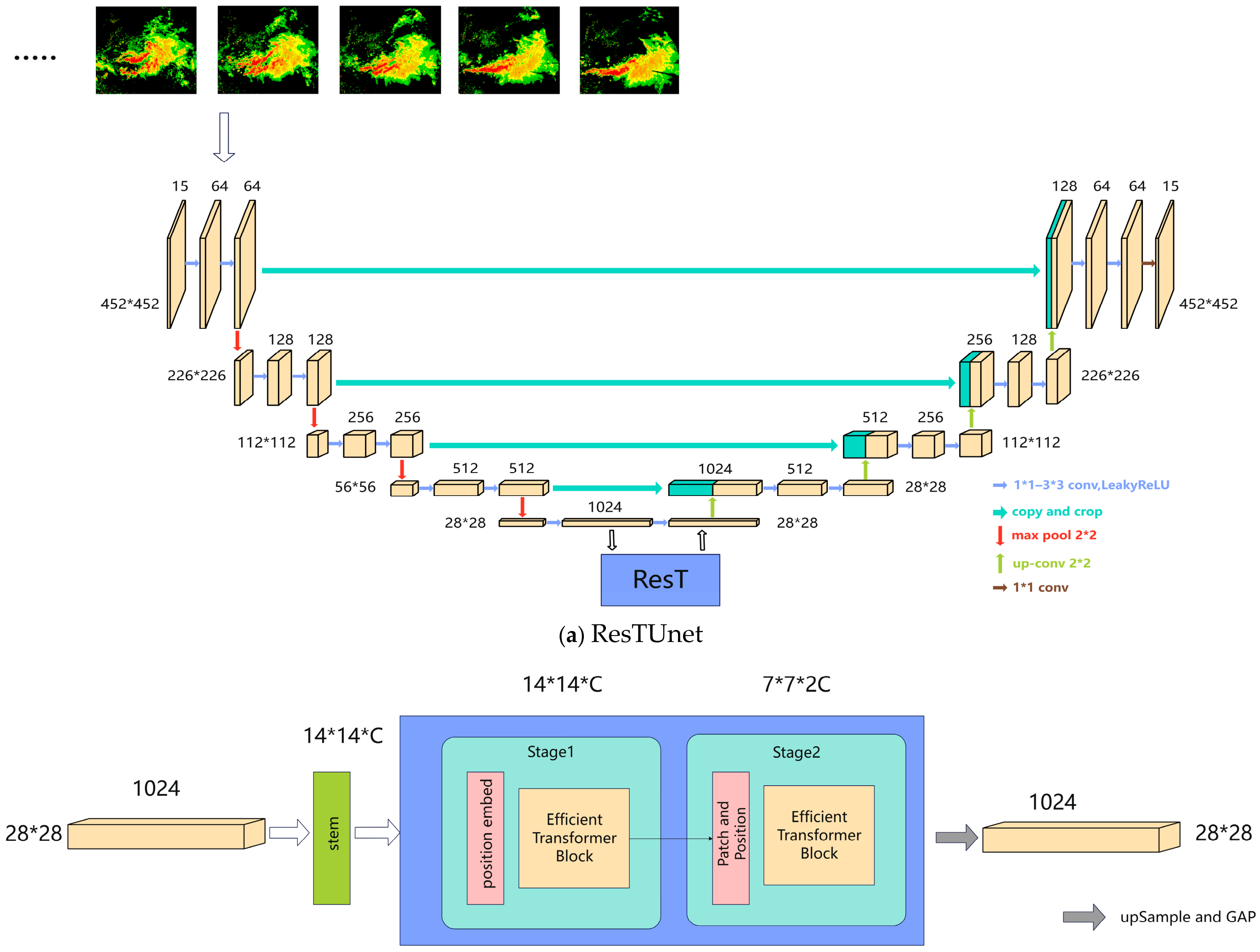
In this paper, we present a deep learning-based approach to extrapolate radar data, which is equivalent to solving a spatio-temporal sequence prediction problem [20]. Our method predicts most-likely future spatial-temporal sequences, $x_1, x_2, x_3, \ldots x_k$ of length k, given a set of spatial-temporal sequences, $x_{-n}, x_{-n+1}, \ldots x_{-1}$ of length n. Each radar echo image x_i at time step i can be viewed as a tensor of height H and width W. To formulate the method, we use

$$argmax\ p(x_1, \ldots, x_k \mid \widehat{x_{-n}}, \ldots, \widehat{x_{-1}}) \tag{1}$$

where $x_1, \ldots .x_k$ represents the predicted sequence of length k, and $\widehat{x_{-\mathrm{n}}}, \ldots, \widehat{x_{-1}}$ denotes the observed history sequence of length n. In our experiments, we aim to forcast radar echo images for the next 90 min, consisting of 15 frames of 6 min each. Each sequence of radar echo images has a height and width of 452 and a sequence length of 15. We use a training set of 30 samples, where every sample includes 15 input data points of radar echo images and 15 actual radar echo images 90 min later.

2.2. ResTUnet

In this paper, we propose ResTUnet, a model based on 1*1 convolution and the ResT network [19] to process a spatiotemporal sequence, as illustrated in Figure 1. ResTUnet adopts an encoder–decoder structure, where the ResT-based encoder has a top-down architecture, whereas the decoder has a bottom-up architecture [20]. The ResT model is a residual attention model akin to Transformer [21] that employs an efficient multi-head self-attention (EMSA) and cross-layer residual connections to extract features. More specifically, the ResT model di-vides input features into multiple heads, using self-attention to compute intra-head correlations. It then enhances feature importance through cross-layer residual connections. The encoder, consisting of convolutional, down sampling, and ResT modules, produces intermediate features while sacrificing some image detail. The decoder restores this loss by directly concatenating the encoder's features. The ResTUnet encoder employs 1*1–3*3 convolutions to minimize computational load by reducing channel dimensions.



(**a**) ResTUnet

(**b**) ResT Model

Figure 1. The overall framework of the ResTUnet Model and the internal details of ResT: Subfigure (**a**) shows the overall framework of ResTUnet, where the 28*28*1024 feature matrix generated by the model during the encoding phase is fed into the ResT module before decoding, where the blue square identifies the exact location of the ResT module. Subfigure (**b**) depicts the working details of the ResT module.

2.3. ResT

The ResT model, inspired by Transformer and ResNet, has four stages, including a stem module for low-level features, position encoding for spatial awareness, and Efficient Transformer blocks for Multi-Head Self-Attention and global capture. We reduced the number of stages in ResT from four to two, utilizing U-net's feature extraction, thereby lowering the parameter count and quickening training without compromising results, as depicted in Figure 1b.

Initially, the input matrix is processed through a patch embedding module to lower its resolution and expand channel dimensions. This is integrated with a position encoding module to refine feature extraction capabilities. To harmonize ResT with Unet, the stem layer in ResT is adjusted to a 3*3 convolution with a stride of 2, negating the need for multiple strides, as Unet already captures low-level features. At each stage's onset, the feature matrix is segmented into spatiotemporal blocks through patch embedding, which consists of a 3*3 convolution with stride 2 and a ReLU activation. Each block undergoes positional encoding, determined through a 3*3 depth convolution (DWConv) and scaled using a sigmoid function, as depicted in Formula (2).

$$\hat{x} = x * \sigma(DWConv(x)) \tag{2}$$

In this formulation, x and $\hat{x}$ represent the input and positionally encoded feature blocks, respectively. The matrix $\hat{x}$ has n rows and c columns for spatial dimensions and channels. The Sigmoid function $\sigma(\cdot)$ scales positional weights within 0–1. $DWConv(\cdot)$ performs depth-wise convolution for spatial location weights, and $*$ signifies the element-wise multiplication of these weights with x. The purpose of positional encoding is to add time and spatial position information to different spatiotemporal feature blocks, enabling the model to better capture global spatiotemporal features.

The input matrix, after patch embedding and position encoding, is fed into an Efficient Transformer block for faster computations. Transformer adopts an encoder–decoder architecture and utilizes the Multi-Head Self-Attention (MSA) mechanism to perform multiple self-attention operations in parallel. Specifically, each head performs a self-attention operation, which can be understood as multiple feature subspaces. This enables the model to capture diverse dependencies within the input sequence across different subspaces. ResT effectively improves the MSA mechanism in Transformer by introducing an Efficient Multi-Head Self-Attention (EMSA) mechanism to optimize computational cost and performance. Three projection transformations, each with k linear layers, map the input matrix from d_m to d_k dimensions, yielding Query (Q), Key (K), and Value (V) matrices. This is represented by Formula (3).

$$EMSA(Q,K,V) = IN\left(Softmax\left(Conv\left(\frac{QK^T}{\sqrt{d_k}}\right)\right)\right)V \tag{3}$$

where $Softmax(S_{ij}) = \frac{e^{S_{ij}}}{\sum_{k=1}^{n} e^{S_{ik}}}$.

And in which $Conv(\cdot)$ denotes a standard 1*1 convolution operation that models interaction between different heads. V is the Value matrix for computing the weighted output sum. Q is the Query matrix matched with the Key matrix K to generate attention weights. $\sqrt{\mathrm{d_k}}$ scales these weights, where $\mathrm{d_k}$ is K's dimensionality. K^T is K's transpose. $EMSA(Q,K,V)$ denotes the Efficient Multi-Head Self-Attention mechanism that processes Q,K,V for self-attention. The $Softmax(\cdot)$ function, also known as the normalized exponential function, is used to compute the weight matrix. S denotes the result of the 1*1 convolution operation, S_{ij} is the element in the j-th column of the i-th row, and $\sum_{k=1}^{n} e^{S_{ik}}$ denotes the exponential sum of all the elements in the i-th row. Therefore, each head's attention function can rely on all keys and queries. However, this impairs MSA's (Multi-Head Self-Attention) [21] ability to jointly focus on information from different representation subsets at different positions. To restore this diversity ability, we added instance normalization [22] (namely $IN(\cdot)$) for the dot-product matrix (after $Softmax(\cdot)$).

Based on the EMSA mechanism described above, the ResT model implements the Efficient Transformer Block, which leverages the efficient attention mechanism of EMSA to capture global information while enhancing computational efficiency. The specific computational procedure of the block is outlined as follows:

$$y = x' + FFN\big(LN(x')\big), \text{and } x' = x + EMSA(LN(x)) \tag{4}$$

$$FFN = \sigma(xW_1 + b_1)W_2 + b_2 \tag{5}$$

After processing the input matrix x based on EMSA, it is subjected to residual linking and then fed into the FFN (Feedforward Neural Network) for feature extraction and nonlinear transformation. The FFN consists of two linear layers and a nonlinear activation function, where $\sigma(\cdot)$ denotes the activation function, which, in this case, is the Gaussian Error Linear Unit (namely $GELU(\cdot)$). Layer normalization (namely $LN(\cdot)$) is applied.

During radar echo extrapolation with the ResTUnet model, ResT is positioned between the Unet encoding and decoding layers. Based on EMSA's efficient feature extraction mechanism, ResT can then process information from the Unet encoding layer and learn the radar spatiotemporal sequence's global features. This ability allows ResT to mitigate the problem of remarkably declining prediction accuracy, which is a challenge for traditional radar extrapolation methods as time progresses.

2.4. 1*1 Convolution

The convolution operation performs a weighted summation of the pixel values within its coverage area by means of a convolution kernel. The convolution kernel slides over the image according to a set step size, and, for an input image of size $H * W * C$, the convolution kernel should have C channels corresponding to each channel of the input image to be processed. The number of convolution kernels determines the number of channels in the output image, i.e., each convolution kernel generates one output channel after processing the input image. The common convolution kernels are 1*1, 3*3, 5*5, etc.

The concept of using 1*1 convolutions, as described, pertains to a convolutional operation with a kernel size of 1*1 [23]. This approach is significant in altering the depth and the number of feature maps within a convolutional layer. A key benefit of this method is its ability to substantially diminish both the computational load and the quantity of parameters required. When incorporated into the convolution operations of a ResTUnet architecture, 1*1 convolutions serve to efficiently decrease the network's computational demands and the total number of parameters. Concurrently, it boosts the number of feature maps. This enhancement in feature maps plays a crucial role in improving the network's capacity to learn and discern more intricate features.

More specifically, the application of a 1*1 convolution occurs prior to a 3*3 convolution. This sequence is instrumental in halving the channel number of the input matrix. For instance, without the use of a 1*1 convolution, transforming a feature matrix from a dimension of h*w*512 to h*w*1024 would necessitate a total of 3*3*512*1024 = 4,718,592 parameters for the 3*3 convolutional operation. However, the introduction of the 1*1 convolutional layer leads to a significant reduction in this parameter count. The new calculation, 1*1*512*256 + 3*3*256*1024, amounts to 2,490,368 parameters. This figure represents a reduction of approximately half the original parameter count, as illustrated in Figure 2. Moreover, the inclusion of 1*1 convolutions contributes to an increase in the depth of the network. This increased depth is integral to enhancing the network's proficiency in capturing and processing a wider array of features.

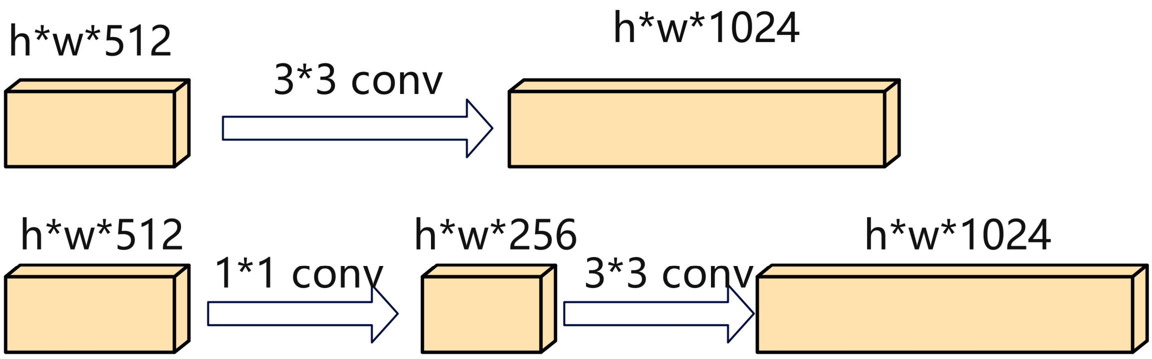


Figure 2. 1*1 convolution.

2.5. Model Conclusion

Our model incorporates the ResT structure, strategically designed to effectively capture and interpret global features in radar spatial-temporal sequences. This design choice directly addresses a common shortcoming in traditional radar extrapolation methods: the significant and rapid deterioration in prediction accuracy over time. The ResT structure helps our model maintain high accuracy levels for extended periods, countering this trend. To enhance our model's capabilities further, we have concentrated on increasing its depth. A key method for achieving this is the strategic employment of 1*1 convolutions. This decision plays a dual role in the model's design. Primarily, it dramatically decreases the total number of parameters that our model requires by approximately half. This reduction goes beyond mere efficiency; it facilitates a more streamlined model that is easier to train and less susceptible to overfitting. Moreover, integrating 1*1 convolutions significantly lowers the computational demand. This is particularly vital, as it enables faster and more efficient data processing, crucial for real-world applications where speed is paramount. A reduced computational load also makes our model more adaptable to various hardware, enhancing its responsiveness and operational speed.

In summary, based on the feature extraction capability of the U-Net model, we synthesize the ResT structure and 1*1 convolution for optimization to form the core architecture. This combination not only optimizes the network's depth but also balances parameter count and computational efficiency. As a result, the model excels in accuracy and efficiency, proving itself to be adaptable and robust across a broad spectrum of radar data analysis and prediction applications.

3. Experiments and Results

3.1. Observed Data

This study utilized 20,000 radar echo images obtained from Zhengzhou East Station Airport between July and August 2022, with a 1024 × 1012 original image resolution. The radar used operates at a wavelength of 5.5 cm, with a raw spatial resolution of 150 m, and a spatial resolution of 1 km based on data processing. The average reflectivity in the range of 10–15 km is considered. Nonetheless, we cropped the images to 452 × 452 pixels because the echo data mainly concentrated on the image's center. Each radar echo image has a time resolution of 6 min/frame. In the data preprocessing stage, we transformed RGB-pixel radar echo images into pixel values that represent echo intensity, ranging from 0 to 80, normalized them to a range of 0 to 1, and then selected 15,000 images with precipitation events. This means that the total number of images available for the experiments in this study is 15,000, 10,500 images for model training, 3000 images for model validation, and 1500 images for model testing. We subdivided these images into 30 sets containing 15 images for training and 15 images for label purposes.

3.2. Evaluation Method

To evaluate model performance under varying rainfall intensities (R), we employ the POD, FAR, CSI, and HSS indicators [24]. In addition, we use the metric Accuracy (ACC) for the classification task to measure the proportion of samples correctly classified by the model in the prediction, and it is worth noting that the same metric is evaluated in the meteorological domain, often referred to as PC (Proportion Correct) [25]. The reflectivity factor (Z) indicates the echo strength from atmospheric particles like water droplets, and it is commonly used in radar weather monitoring to signify precipitation characteristics. However, the relationship between the reflectivity factor and rainfall intensity is not a one-to-one correspondence because the reflectivity factor is not only related to the number of raindrops but also affected by factors such as particle size, shape, and density [11]. Therefore, it is necessary to use the Z-R relationship [26] $log_{10}R = \frac{Z-10*log_{10}a}{10*b}$ (where a and b are statistical constants, 58.53 and 1.56, respectively [12]) to convert the reflectivity factor Z into the specific rainfall intensity value R. Table 1 shows the rainfall intensity R and the corresponding radar reflectivity factor from light rain to heavy rain. For a designated threshold R, corresponding to a radar reflectivity k, we analyze both the predicted and the true radar echo images. Each pixel p_{ij} is assigned either 0 or 1 depending on whether it is below or above the value k. We compute counts for four categories: hit pixels n_{hits}(predicted $p_{ij} = 1$, true $p_{ij} = 1$), miss pixels n_{miss}(predicted $p_{ij} = 0$, true $p_{ij} = 1$), false pixels n_{false}(predicted $p_{ij} = 1$, true $p_{ij} = 0$), and background pixels n_{bd}(predicted $p_{ij} = 0$, true $p_{ij} = 0$). Based on these counts, the definitions of the metrics POD, FAR, CSI, HSS, and ACC are as follows [24].

$$POD = \frac{n_{hits}}{n_{hits} + n_{miss}} \tag{6}$$

$$FAR = \frac{n_{false}}{n_{hits} + n_{false}} \tag{7}$$

$$CSI = \frac{n_{hits}}{n_{hits} + n_{miss} + n_{false}} \tag{8}$$

$$HSS = \frac{2 * \left(n_{hits} * n_{bd} - n_{miss} * n_{false}\right)}{(n_{hits} + n_{miss})(n_{miss} + n_{bd})\left(n_{hits} + n_{false}\right)\left(n_{false} + n_{bd}\right)} \tag{9}$$

$$ACC = \frac{n_{hits} + n_{bd}}{n_{hits} + n_{miss} + n_{false} + n_{bd}} \tag{10}$$

Table 1. Rainfall intensity and reflectivity factor under different rainfall levels.

Rainfall Level	Rainfall Intensity	Reflectivity Factor
Light to moderate	$2 \leq$ R(mm/h) < 5	$22.3 \leq$ Z(dBZ) < 28.5
Moderate	$5 \leq$ R(mm/h) < 10	$28.5 \leq$ Z (dBZ) < 33.2
Moderate to heavy	$10 \leq$ R(mm/h) < 30	$33.2 \leq$ Z (dBZ) < 40.7
Rainstorm warning	$30 \leq$ R(mm/h)	$40.7 \leq$ Z(dBZ)

3.3. Experiment Settings

In our experiments, we utilized the ResTUnet architecture with a 0.0001 initial learning rate and a batch size of 1. The model was optimized using a combined MSE and L1 loss function [27] for enhanced robustness and overfitting resistance. This MSE-L1 trade-off facilitated a thorough performance evaluation under various conditions. All models were trained on a NVIDIA 3090 GPU using the ADAM optimizer [28], with input images normalized to [0, 1]. Additionally, the same training settings were applied to other models beyond ResTUnet. We used the official implementations of PredRNN++ (available at https://github.com/Yunbo426/predrnn-pp, accessed on 1 November 2022). For the other models, we adopted the best available implementations from GitHub.

3.4. Experimental Results

To provide a more intuitive comparison among different methods, we generated a visual representation of the radar extrapolation outcomes achieved by each method in Figure 3. From this, it is apparent that our proposed ResTUnet method yields more precise predictions for both shape and intensity as time continues. Conversely, other models exhibit significant differences between their shape and intensity predictions and the ground truth values over time progression. From Figure 3, it can be seen that our model outperforms other models in terms of the shape and intensity of radar echoes. This indicates that applying the ResT structure to the Unet model is effective. In addition, the radar echo images generated by our model maintain similar shape and intensity to the ground truth even after 60 min. Compared to our model, convLSTM preserves the regions of high-intensity radar echoes but almost no regions of low-intensity radar echoes. This may be because the network structure of convLSTM is more suitable for capturing and representing the features of strong radar echo regions. Strong radar echo regions usually have more obvious edges, textures, and structures, which are features that can be easily learned by convLSTM models [29]. The ConvGRU preserves a higher degree of weak radar echo regions. This may be because the gating mechanism of GRUs is more suitable for processing the data patterns of weak echoes. However, it is not effective in handling the data patterns of strong radar echo regions [30].

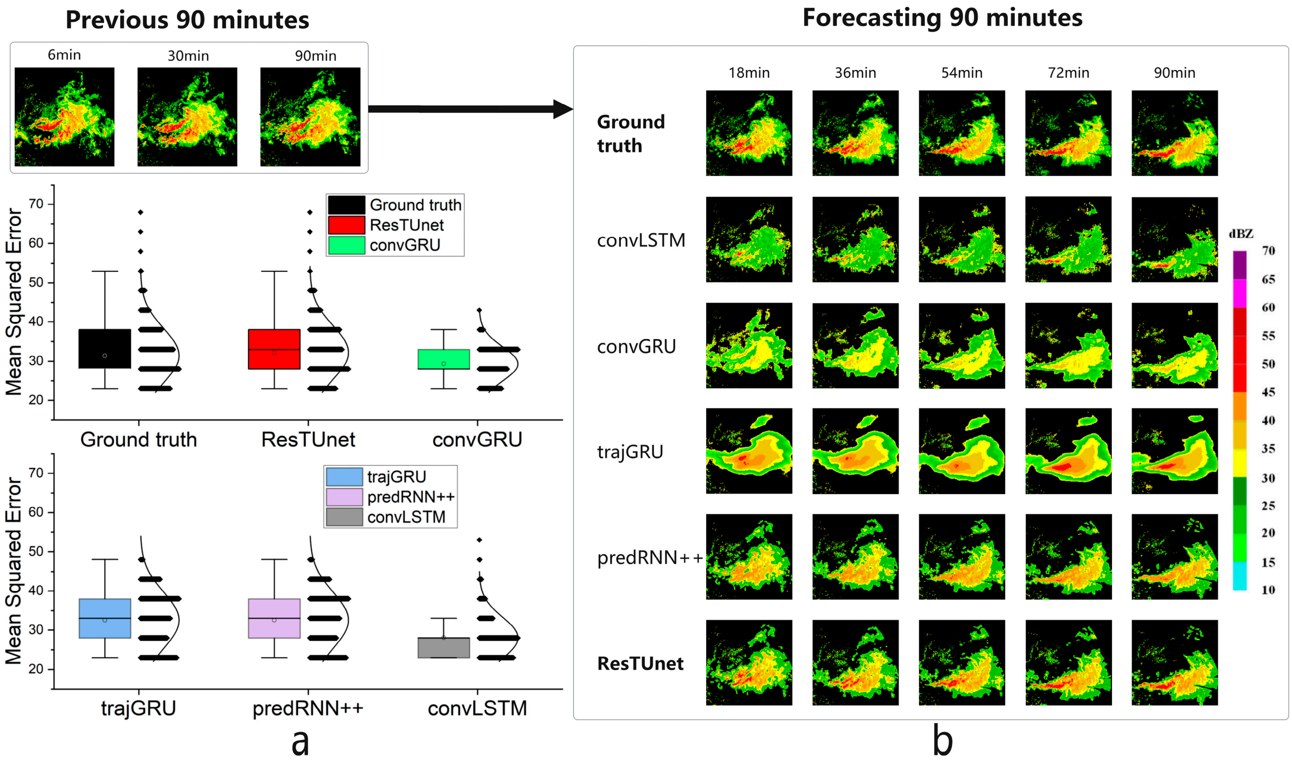


Figure 3. Predicted results under different radar extrapolation methods: (**a**) Comparison of the predicted results range under different radar extrapolation methods (**bottom left**). (**b**) Visualization of predicted results under different radar extrapolation methods (**right**).

The TrajGRU has higher accuracy in predicting radar echo intensity, but the details of generating radar echo shapes are not clear enough. This may be because the TrajGRU is more suitable for modeling and predicting the intensity of radar echoes. Terrain shapes usually have more complex and fine-grained features, and the TrajGRU may need deeper modeling capabilities to accurately represent and generate these details [31]. The PredRNN++ performs well in terms of shape and intensity, but accuracy still needs improvement. From Figure 3, it can be seen that, compared to other models, the distribution of radar echoes generated by our model is closest to the real distribution. Tables 2 and 3 present the outcomes of several approaches, which include ConvLSTM, the ConvGRU, the TrajGRU, the PredRNN+, and our proposed ResTUnet model. To better compare the performance differences, we independently calculated the values for the CSI, HSS, POD, and FAR. The presented data show that ResTUnet achieved the highest performance in almost all assessments. Specifically, when the rainfall threshold equaled 30 mm/h, ResTUnet exhibited a 17.20% higher CSI, 11.97% higher HSS, and 11.35% higher POD than the second-best performing model, which was the PredRNN++. Additionally, compared to the second-best TrajGRU, ResTUnet had a 10.06% lower FAR. These results suggest that including the ResT structure in the Unet architecture can effectively capture the global spatiotemporal characteristics of radar echoes, which ultimately helps overcome the challenge of rapid deterioration of the forecasting accuracy of radar extrapolation techniques with time progression.

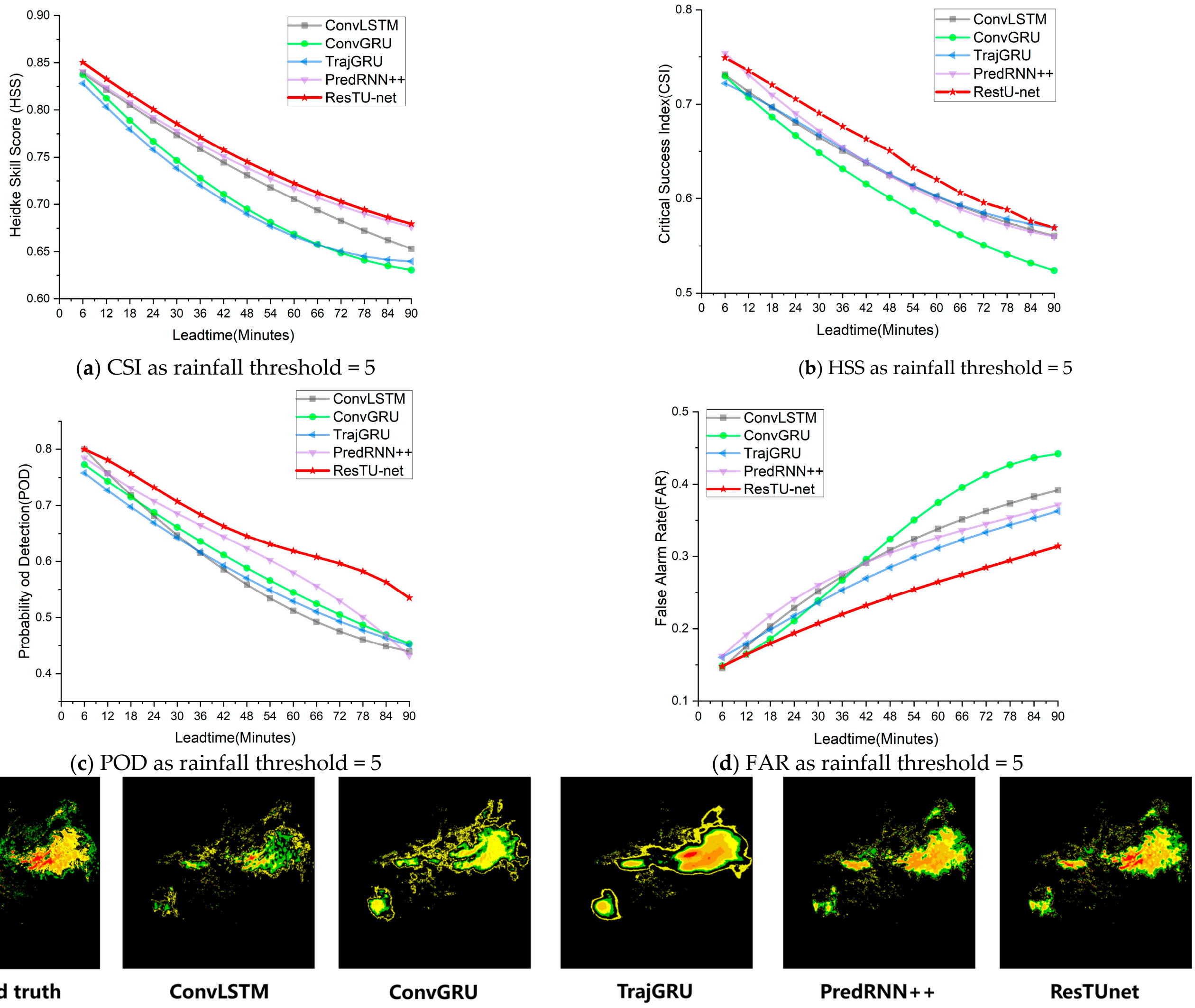
Table 2. Compare the results of the 2020 Zhengzhou Airport Doppler radar observation dataset in terms of the Critical Success Index (CSI) and Heidke Skill Score (HSS). Bold indicates the best evaluation index among all models. An upward arrow (↑) indicates that the performance of the model improves as the indicator value increases.

Rainfall Threshold (mm/h)	**CSI↑**			**HSS↑**		
	5	**10**	**30**	**5**	**10**	**30**
ConvLSTM	0.7136	0.5467	0.1879	0.8217	0.6515	0.2918
ConvGRU	0.7076	0.5779	0.1976	0.8125	0.6194	0.2784
TrajGRU	0.7105	0.5710	0.2150	0.8030	0.6325	0.3058
PredRNN++	0.7313	0.6204	0.2488	0.8240	0.6841	0.3491
ResTUnet	**0.7334**	**0.6345**	**0.2916**	**0.8329**	**0.7033**	**0.3909**

Table 3. Compare the results of the 2020 Zhengzhou Airport Doppler radar observation dataset in terms of the Probability of Detection (POD) and False Alarm Rate (FAR). Bold indicates the best evaluation index among all models. An upward arrow (↑) indicates that the performance of the model improves as the indicator value increases. Conversely, a downward arrow (↓) indicates the opposite, meaning that as the value of the metric decreases, the model performs better.

Rainfall Threshold (mm/h)	**POD↑**			**FAR↓**		
	5	**10**	**30**	**5**	**10**	**30**
ConvLSTM	0.7578	0.5946	0.2461	0.1755	0.3257	0.6283
ConvGRU	0.7435	0.6134	0.2774	**0.1645**	0.3475	0.6439
TrajGRU	0.7270	0.6301	0.2646	0.1793	0.3098	0.5906
PredRNN++	0.7567	0.6555	0.2976	0.1918	0.2889	0.6001
ResTUnet	**0.7769**	**0.6702**	**0.3312**	0.1682	**0.2751**	**0.5312**

In order to compare the results of different methods, we plotted the CSI, HSS, POD, and FAR curves of all methods within the 0–90 min time range. From Figure 4, it can be seen that, for a rainfall threshold of five, our model improved by 0.29%, 1.08%, and 2.52% for the CSI, HSS, and POD, respectively, compared to the second-ranked model. From Figure 5, it can be seen that, for a rainfall threshold of 10, our model improved by 2.27%, 2.81%, and 2.24% for the CSI, HSS, and POD, respectively, compared to the second-ranked model, while it decreased by 4.78% for the FAR. Finally, from Figure 6, it can be seen that, for a rainfall threshold of 30, our model improved by 17.20%, 11.97%, and 11.36% for the CSI, HSS, and POD, respectively, compared to the second-ranked model, while it decreased by 10.06% for the FAR. Specifically, using the experimental results in Tables 2 and 3, we calculated the difference in experimental results between the optimal model ResTUnet and the second-best model for different metrics and different rainfall thresholds to determine the percentage improvement in extrapolation performance of ResTUnet relative to that model. For instance, to determine the 17.20% improvement in the CSI metrics for ResTUnet compared to the PredRNN++, we first calculated a 4.28% enhancement in the CSI metrics for ResTUnet. We then express this improvement (4.28%) as a percentage of the CSI result from the PredRNN++ (24.88%), which leads to an overall improvement of 17.20%.



(**a**) CSI as rainfall threshold = 5

(**b**) HSS as rainfall threshold = 5

(**c**) POD as rainfall threshold = 5

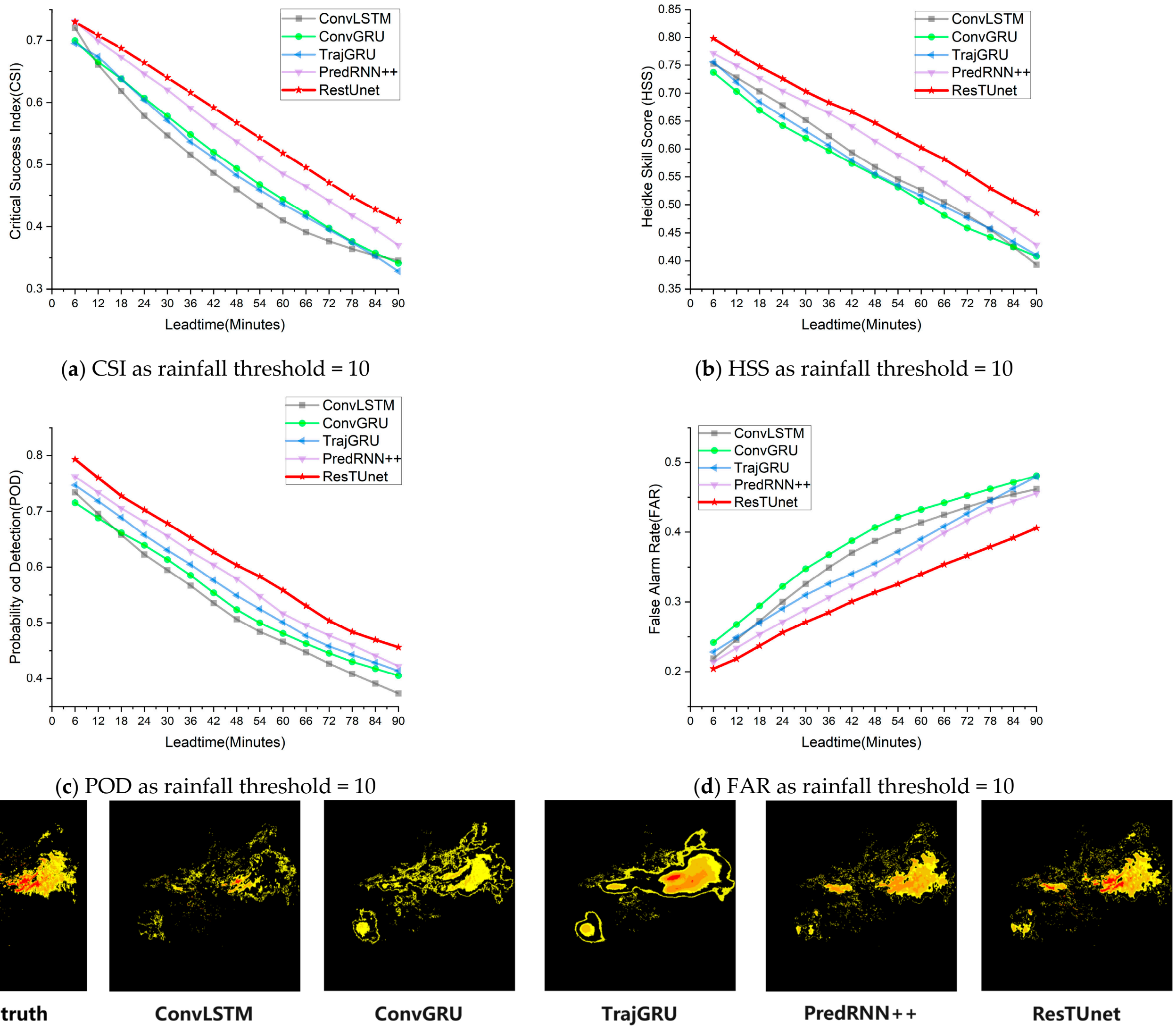
(**d**) FAR as rainfall threshold = 5

(**e**) Visualization of predicted results under different radar extrapolation methods for rainfall threshold = 5

Figure 4. Comparison of CSI, HSS, POD, and FAR curves as rainfall threshold = 5.

In addition, Figure 7 shows the change in the mean square error of the ResTUnet model over time under different rainfall thresholds. This indicates that, in the cases where R = 0 and R = 5, the mean square error remains stable within the 0–60 min period (about 15%), while, in the case of R = 10, it remains stable during this time frame (about 17%). In the case of R = 30, the error remains relatively stable in the 0–60 min period (about 23%). In all cases, the mean square error starts to increase only after 60–90 min. Overall, our model exhibits the most gradual changes across all curves, which suggests that our proposed approach successfully captures the global features of the radar echo spatiotemporal sequence and addresses the issue of fast decline in the prediction accuracy of radar extrapolation techniques as time increases.

Figure 4 shows the comparison of various indicators between our model and other models with a rainfall threshold of five. From Figure 4, it can be seen that our model's indicators at each time step are better than those of other models. Specifically, in terms of the False Alarm Rate (FAR) indicator, the curve of our model is flatter than that of other models, indicating that our model is better at capturing global spatial-temporal features than other models. In Figure 6, when the rainfall threshold is 30, our model's indicators are even better, indicating that our model has a stronger ability to learn high-intensity radar echo features than other models. Figure 7 shows the distribution and frequency of mean squared errors of our model at different thresholds. It can be seen from Figure 7 that the mean squared error of our model remains stable within 0–60 min, while there is a large change between 60–90 min. This indicates that the ResT structure in our model can indeed learn global spatial-temporal features. It can be seen from Figure 7a that our model performs relatively stable when the rainfall threshold is 0, and there are fewer outliers at each time step. This may be because our model more accurately predicts the areas where radar reflectivity is reduced. In contrast, we observed in Figure 7b that the accuracy of our model decreases when the rainfall threshold is five, while it is highest when the rainfall threshold is zero. In Figure 7d, the prediction accuracy decreases more significantly when the rainfall threshold is 30. Nevertheless, relative to other models, our model still exhibits superior prediction accuracy.



(**a**) CSI as rainfall threshold = 10

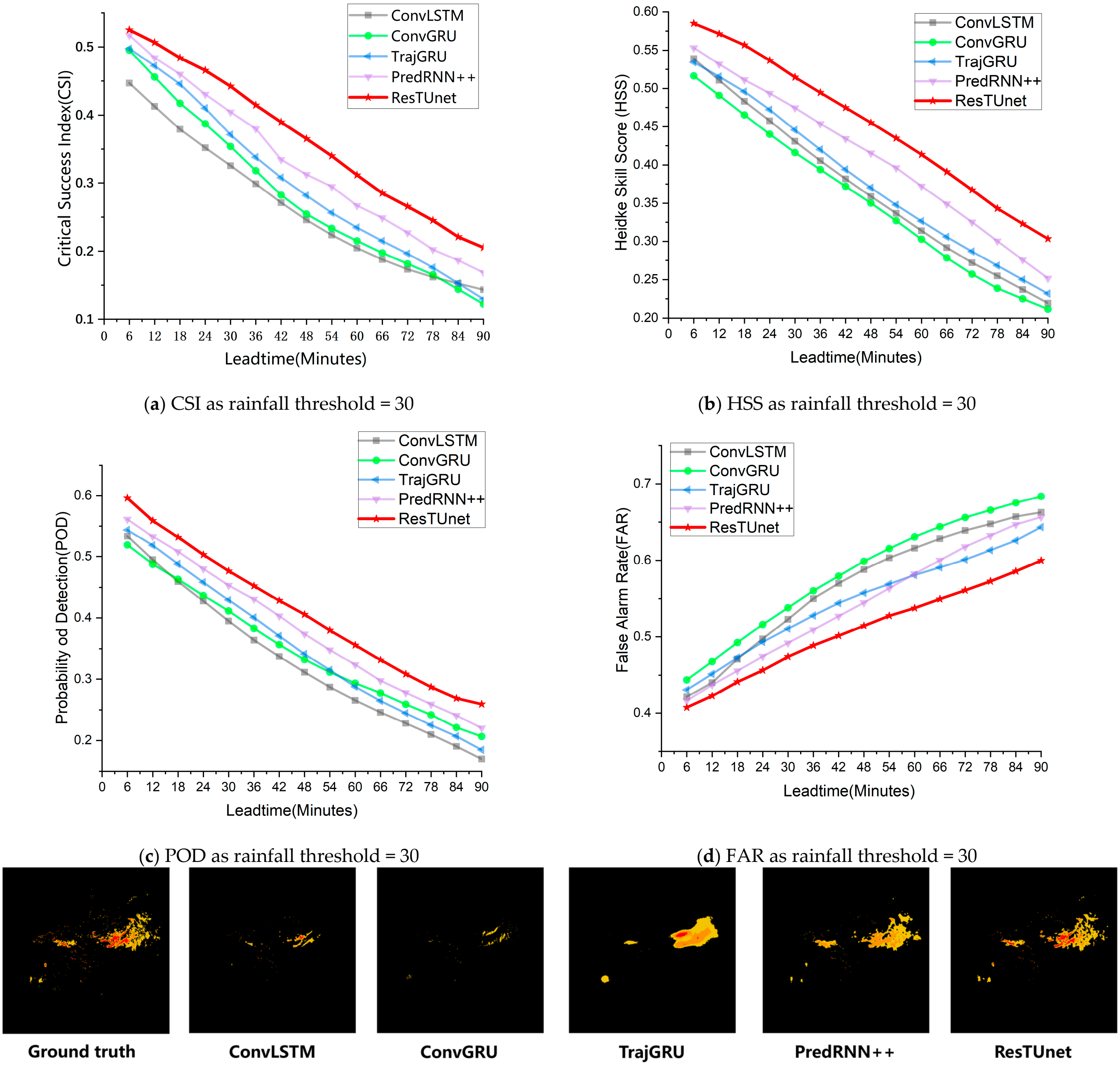
(**b**) HSS as rainfall threshold = 10

(**c**) POD as rainfall threshold = 10

(**d**) FAR as rainfall threshold = 10

(**e**) Visualization of predicted results under different radar extrapolation methods for rainfall threshold = 10

Figure 5. Comparison of CSI, HSS, POD, and FAR curves as rainfall threshold = 10.



(**a**) CSI as rainfall threshold = 30

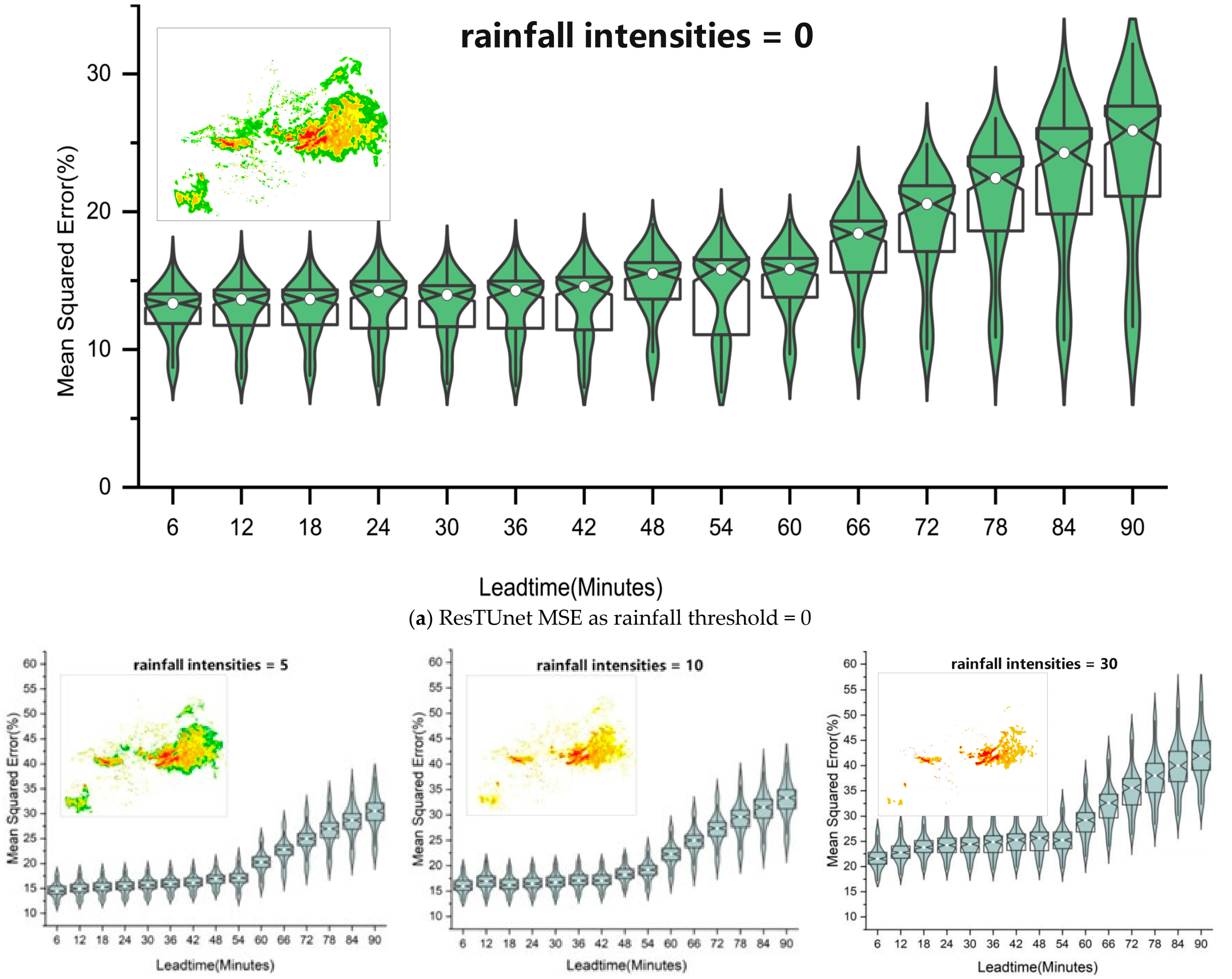
(**b**) HSS as rainfall threshold = 30

(**c**) POD as rainfall threshold = 30

(**d**) FAR as rainfall threshold = 30

(**e**) Visualization of predicted results under different radar extrapolation methods for rainfall threshold = 30

Figure 6. Comparison of CSI, HSS, POD, and FAR curves as rainfall threshold = 30.



(**a**) ResTUnet MSE as rainfall threshold = 0

(**b**) ResTUnet MSE as rainfall threshold = 5(**c**) ResTUnet MSE as rainfall threshold = 10(**d**) ResTUnet MSE as rainfall threshold = 30

Figure 7. ResTUnet MSE changes over time.

3.5. Ablation Study

In our comprehensive study, we conducted an in-depth ablation analysis of the ResTUnet architecture to precisely determine the impact and significance of its individual components, specifically focusing on the ResT and 1*1 convolution features. This analysis was essential to understand how each module contributes to the overall performance and efficiency of the model. We meticulously designed four distinct variants of the ResTUnet model for this purpose: the first variant was ResTUnet stripped of both the 1*1 convolution and ResT module, essentially excluding these key features; the second variant retained the ResT module but did not include the 1*1 convolution; the third variant kept the 1*1 convolution but excluded the ResT module; and the fourth was the complete ResTUnet model.

To evaluate the performance of each variant, we employed a comprehensive set of metrics. We measured the model parameter count in millions (M), which gave us insight into the complexity and scalability of each variant. The computational speed was assessed in FLOPs (Floating Point Operations Per Second), providing a clear picture of the efficiency and responsiveness of the models in practical applications. Additionally, we utilized the CSI (Critical Success Index) and ACC (Accuracy) metrics for a more nuanced understanding of model performance, particularly focusing on scenarios with zero rainfall intensity, which are challenging for radar prediction models. The results, as depicted in Table 4, were revealing. The absence of either the 1*1 convolution or the ResT module significantly impacted the CSI and ACC scores of the ResTUnet, underscoring the vital role played by the ResT module in radar-related tasks. Notably, the variant without the 1*1 convolution showed results that were closely aligned with those of the full ResTUnet model, thereby confirming the standalone importance of the ResT module in the architecture. On the other hand, the variant lacking the ResT module, while benefitting from quicker inference times and a reduced parameter count due to the presence of the 1*1 convolution, did not perform as well on other metrics.

Table 4. Compare the results of the 2020 Zhengzhou Airport Doppler radar observation dataset in terms of model parameters, inference speed, the Critical Success Index (CSI), and Accuracy (ACC).

Model	Params (M)	FLOPs (G)	CSI	ACC
ResTUnet without 1*1 convolution and ResT	21.32	204.8	0.2645	0.4574
ResTUnet without 1*1 convolution	30.13	298.6	0.5312	0.7458
ResTUnet without ResT	18.85	172.4	0.2726	0.4869
ResTUnet	25.47	198.2	0.5346	0.7512

In summary, this comprehensive ablation study highlighted that, while each component of the ResTUnet architecture has its unique contributions, it is their integration that allows the model to excel. The full ResTUnet model demonstrated superior performance across all metrics, striking an optimal balance between complexity and efficiency and underscoring the effectiveness of combining these two powerful features in radar data analysis and prediction tasks. This study not only validated the design choices made in the ResTUnet architecture but also provided valuable insights for future enhancements and developments in the field.

4. Conclusions

This research introduces ResTUnet, an innovative radar echo extrapolation model that synergistically combines the strengths of the ResT and Unet architectures, along with the strategic use of 1*1 convolution. ResTUnet is designed to address the limitations of traditional radar extrapolation methods, particularly the issue of declining prediction accuracy over time. The Unet architecture, a core component of ResTUnet, features a symmetrical encoding–decoding structure. This structure is pivotal in ensuring efficient information transmission across various layers of the network, thereby significantly enhancing the model's overall accuracy. The encoding–decoding framework effectively captures and reconstructs the radar data, ensuring that crucial spatial-temporal features are retained and accurately represented. Moreover, the implementation of skip connections in the Unet portion of the model serves a critical role. These connections act as bridges, transferring detailed information directly from the encoding layers to the corresponding decoding layers. This transfer helps in preserving the integrity of the original image's details, which is instrumental in improving the prediction accuracy of the model. In parallel, the ResT architecture is meticulously integrated into ResTUnet. It incorporates block embedding and position encoding modules, which are essential for effective feature extraction. These modules enhance the model's ability to accurately identify and process spatial and temporal patterns in radar data, thereby significantly boosting the model's predictive accuracy. The combination of the ResT architecture with the Unet encoding layer is particularly effective in learning the global features of radar spatiotemporal sequences. This integration addresses a critical shortfall in traditional radar extrapolation methods—the rapid decrease in prediction accuracy over time.

The application of 1*1 convolution within ResTUnet marks a strategic advancement in the model's design. By utilizing 1*1 convolution, the model significantly reduces the number of parameters and the computational burden. This reduction not only makes the network more efficient but also allows for faster processing and response times, which are crucial for real-time radar data analysis. Additionally, 1*1 convolution increases the number of feature maps in the network. This increase is vital as it enables the network to learn and represent a broader range of features, further enhancing the model's predictive capabilities. Moreover, the inclusion of 1*1 convolution contributes to an increase in the network's depth. This deeper network architecture leads to more effective feature extraction and overall improved network performance. The depth enables the network to discern more subtle and complex patterns in the data, which is essential for accurate radar echo extrapolation.

In conclusion, our proposed ResTUnet model marks a significant advancement in radar echo extrapolation, surpassing contemporary models like convLSTM, the convGRU, the TrajGRU, and the predRNN++. Our comprehensive experiments, conducted under various conditions and focusing on different rain intensities crucial for weather forecasting, validate ResTUnet's effectiveness. We ensured a fair comparison by using identical datasets of radar echo sequences across a broad spectrum of meteorological scenarios. ResTUnet excelled in accuracy, computational efficiency, handling temporal dynamics, and spatial detail precision, especially in maintaining accuracy across varying rain intensities. Its superior performance is attributed to its advanced architecture, merging ResT and Unet structures with 1*1 convolutions, enabling a more effective global and local radar data feature capture. ResTUnet's computational efficiency, vital for real-time weather forecasting, allows rapid data processing without accuracy compromise, making it an invaluable tool for meteorologists.

Deep learning models like ResTUnet perform well in regular precipitation scenarios with stable distributions, but due to uneven data distribution and low sensitivity to complex features, the extracted features may not be sufficiently accurate when the model is faced with more complex feature spaces and strong spatial-temporal nonlinear relationships. This results in a model that is more accurate in predicting the overall precipitation field but struggles to effectively predict rare and intense precipitation events. Furthermore, the limited number of radar echo images in the available dataset affects the learning ability of the model. In future studies, we expect more data to be available to explore the extrapolation potential of the model in more depth. Even so, we believe that the improved results of the present model have provided important references and guidance for subsequent work on spatio-temporal sequence prediction (e.g., radar echo extrapolation, irradiance prediction based on satellite cloud maps, etc.).

Furthermore, there is still room for improvement in our model. In practical weather forecasting, there are sometimes long-term forecasts exceeding 2 h, which our model cannot accurately predict [32]. This is due to the ResT model's limited ability to capture global characteristics with long-term dependencies. To solve this problem, we can approach it from the perspective of multi-scale modeling, decompose the prediction problem into different time scales, and use appropriate models to model each time scale [33]. For example, we can decompose the long-term prediction task into multiple subtasks, predict the weather conditions within different time ranges separately, and then integrate the results of these subtasks. In the future, we will conduct research to improve the model's performance in this regard.

Author Contributions: Conceptualization, L.Z. (Lei Zhang), R.Z. and Y.W. (Yadong Wang); methodology, Y.W. (Yu Wu), R.Z. and Y.W. (Yadong Wang); software, Z.W., R.Z. and Y.W. (Yadong Wang); validation, Y.W. (Yu Wu) and Y.W. (Yadong Wang).; formal analysis, L.Z. (Lei Zhang); investigation, R.Z. and Y.W. (Yadong Wang); resources, L.Z. (Lei Zhang) and Y.W. (Yu Wu); data curation, Y.W. (Yadong Wang); writing—original draft preparation, Y.W. (Yadong Wang); writing—review and editing, L.Z. (Lijuan Zheng), C.X. and Y.W. (Yadong Wang); visualization, X.Z.; supervision, L.Z. (Lei Zhang) and Y.W. (Yu Wu); project administration, R.Z., Y.W. (Yadong Wang) and Y.Z.; funding acquisition, L.Z. (Lei Zhang), Y.W. (Yu Wu) and L.Z. (Lijuan Zheng). All authors of this paper have directly participated in the planning, execution, or analysis of this study. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Key Research and Development Program of China (Grant 2022YFB3902200); National Natural Science Foundation of China (Grant 42071318); Key R&D Projects in Henan Province (Grant 241111212800); Scientific and Technological Key Project in Henan Province (Grant 232102240020); China University Research Innovation Fund (Grant 2023DT012); and Beijing Engineering Research Center of Aerial Intelligent Remote Sensing Equipments Fund (AIRSE202408).

Data Availability Statement: The authors declare that there are no competing financial interests. The data are available on https://doi.org/10.6084/m9.figshare.25053089, accessed on 24 January 2024.

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

References

1. Reyniers, M. *Quantitative Precipitation Forecasts Based on Radar Observations: Principles, Algorithms and Operational Systems*; Institut Royal Météorologique de Belgique: Brussel, Belgium, 2008.
2. Wilson, J.W.; Crook, N.A.; Mueller, C.K.; Sun, J.; Dixon, M. Nowcasting thunderstorms: A status report. *Bull. Am. Meteorol. Soc.* **1998**, *79*, 2079–2100. [CrossRef]
3. Berenguer, M.; Surcel, M.; Zawadzki, I.; Xue, M.; Kong, F. The diurnal cycle of precipitation from continental radar mosaics and numerical weather prediction models. part ii: Intercomparison among numerical models and with nowcasting. *Mon. Weather Rev.* **2012**, *140*, 2689–2705. [CrossRef]
4. Chen, L.; Cao, Y.; Ma, L.; Zhang, J. A deep learning-based methodology for precipitation nowcasting with radar. *Earth Space Sci.* **2020**, *7*, e2019EA000812. [CrossRef]
5. Rinehart, R.; Garvey, E. Three-dimensional storm motion detection by conventional weather radar. *Nature* **1978**, *273*, 287–289. [CrossRef]
6. Li, L.; Schmid, W.; Joss, J. Nowcasting of motion and growth of precipitation with radar over a complex orography. *J. Appl. Meteorol. Climatol.* **1995**, *34*, 1286–1300. [CrossRef]
7. Mingxuan, C.; Yingchun, W.; Xiaoding, Y. Improvement and application test of trec algorithm for convective storm nowcast. *J. Appl. Meteorol. Sci.* **2007**, *18*, 690–701.
8. Bowler, N.E.; Pierce, C.E.; Seed, A. Development of a precipitation nowcasting algorithm based upon optical flow techniques. *J. Hydrol.* **2004**, *288*, 74–91. [CrossRef]
9. Woo, W.C.; Wong, W.K. Operational application of optical flow techniques to radar-based rainfall nowcasting. *Atmosphere* **2017**, *8*, 48. [CrossRef]
10. Han, L.; Liang, H.; Chen, H.; Zhang, W.; Ge, Y. Convective precipitation nowcasting using u-net model. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 4103508. [CrossRef]
11. 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. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 802–810. [CrossRef]
12. 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. [CrossRef]
13. Wang, Y.; Long, M.; Wang, J.; Gao, Z.; Yu, P.S. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 879–888.
14. 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, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 5123–5132.
15. Sønderby, C.K.; Espeholt, L.; Heek, J.; Dehghani, M.; Oliver, A.; Salimans, T.; Agrawal, S.; Hickey, J.; Kalchbrenner, N. Metnet: A neural weather model for precipitation forecasting. *arXiv* **2020**, arXiv:2003.12140.
16. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Proceedings of the 18th International Conference, Munich, Germany, 5–9 October 2015*; Part III 18; Springer: Munich, Germany, 2015; pp. 234–241.
17. Trebing, K.; Staczyk, T.; Mehrkanoon, S. Smaat-unet: Precipitation nowcasting using a small attention-unet architecture. *Pattern Recognit. Lett.* **2021**, *145*, 178–186. [CrossRef]
18. Espeholt, L.; Agrawal, S.; Sønderby, C.; Kumar, M.; Heek, J.; Bromberg, C.; Gazen, C.; Carver, R.; Andrychowicz, M.; Hickey, J.; et al. Deep learning for twelve hour precipitation forecasts. *Nat. Commun.* **2022**, *13*, 5145. [CrossRef] [PubMed]
19. Zhang, Q.; Yang, Y.B. Rest: An efficient transformer for visual recognition. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 15475–15485.
20. Weyn, J.A.; Durran, D.R.; Caruana, R. Can machines learn to predict weather? using deep learning to predict gridded 500-hpa geopotential height from historical weather data. *J. Adv. Model. Earth Syst.* **2019**, *11*, 2680–2693. [CrossRef]
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010. [CrossRef]
22. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Instance normalization: The missing ingredient for fast stylization. *arXiv* **2016**, arXiv:1607.08022.

23. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
24. Schaefer, J.T. The critical success index as an indicator of warning skill. *Weather Forecast.* **1990**, *5*, 570–575. [CrossRef]
25. Thornes, J.E.; Stephenson, D.B. How to judge the quality and value of weather forecast products. *Meteorol. Appl.* **2001**, *8*, 307–314. [CrossRef]
26. PLi, W.; Wong, W.K.; Chan, K.Y.; Lai, E.S.T. *SWIRLS—An Evolving Nowcasting System*; Hong Kong Special Administrative Region Government: Hong Kong, China, 2000.
27. Chai, T.; Draxler, R.R. Root mean square error (rmse) or mean absolute error (mae). *Geosci. Model Dev. Discuss.* **2014**, *7*, 1525–1534.
28. Zhang, Z. Improved adam optimizer for deep neural networks. In Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018; pp. 1–2.
29. Wang, D.; Yang, Y.; Ning, S. Deepstcl: A deep spatiotemporal convlstm for travel demand prediction. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
30. Tian, L.; Li, X.; Ye, Y.; Xie, P.; Li, Y. A generative adversarial gated recurrent unit model for precipitation nowcasting. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 601–605. [CrossRef]
31. Zhang, L.; Huang, Z.; Liu, W.; Guo, Z.; Zhang, Z. Weather radar echo prediction method based on convolution neural network and long short-term memory networks for sustainable e-agriculture. *J. Clean. Prod.* **2021**, *298*, 126776. [CrossRef]
32. Xiang, Z.; Demir, I. Distributed long-term hourly streamflow predictions using deep learning–a case study for state of iowa. *Environ. Model. Softw.* **2020**, *131*, 104761. [CrossRef]
33. Cui, Z.; Chen, W.; Chen, Y. Multi-scale convolutional neural networks for time series classification. *arXiv* **2016**, arXiv:1603.06995.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.