








SpatioTemporal Inference Network for Precipitation Nowcasting With Multimodal Fusion

Qizhao Jin , Xinbang Zhang , Xinyu Xiao , Ying Wang , Gaofeng Meng , *Senior Member, IEEE*, Shiming Xiang , *Member, IEEE*, and Chunhong Pan , *Member, IEEE*

Abstract—Precipitation plays a significant role in global water and energy cycles, largely affecting many aspects of human life, such as transportation and agriculture. Recently, meteorologists have tried to predict precipitation with deep learning methods by learning from much historical meteorological data. Under this paradigm, the task of precipitation nowcasting is formulated as a spatiotemporal sequence forecasting problem. However, current studies suffer from two inherent drawbacks of the definition of the problem. First, considering that the weather patterns vary in spatial and temporal dimensions, a spatiotemporally shared kernel is not optimal for capturing features across different regions and seasons. Second, these methods isolate the precipitation from other meteorological elements, such as temperature, humidity, and wind. The disability of cross-model learning prevents the possibility of the promotion of precipitation prediction. Therefore, this article proposes a spatiotemporal inference network (STIN) to produce precipitation prediction from multimodal meteorological data with spatiotemporal specific filters. Specifically, we first design a spatiotemporal-aware convolutional layer (STAConv), in which kernels are generated conditioned on the incoming spatiotemporally features vector. Replacing normal convolution with STAConv enables the extraction of spatiotemporal specific information from the meteorological data. Based on the STAConv, the spatiotemporal-aware convolutional neural network (STACNN) is further proposed, fusing the multimodal information, including temperature, humidity, and wind. Then, an encoder–decoder framework composed of RNN layers is built to extract representative temporal dynamics from multimodal information. To investigate the practicality of the proposed method, we employ STIN to predict the following precipitation intensity. Extensive experiments on three meteorological datasets demonstrate the effectiveness of our model on precipitation nowcasting.

Index Terms—Data mining, multimodal knowledge discovery, precipitation nowcasting.

Manuscript received 22 February 2023; revised 10 June 2023 and 3 September 2023; accepted 20 September 2023. Date of publication 13 October 2023; date of current version 14 December 2023. This work was supported by the National Natural Science Foundation of China under Grant 62076242. (*Corresponding author: Xinyu Xiao.*)

Qizhao Jin, Xinbang Zhang, and Xinyu Xiao are with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: qizhao.jin@nlpr.ia.ac.cn; xinbang.zhang@nlpr.ia.ac.cn; xinyu.xiao@nlpr.ia.ac.cn).

Ying Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: ywang@nlpr.ia.ac.cn; gfmeng@nlpr.ia.ac.cn; smxiang@nlpr.ia.ac.cn; chpan@nlpr.ia.ac.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3321963

I. INTRODUCTION

PRECIPITATION nowcasting is forecasting rainfall intensity in the short term up to a few hours. It is conditioned on the known meteorological elements collected by radar, satellite, and surface weather meteorological stations. Accurate prediction is crucial for the production and people’s livelihood, including agriculture planting, energy management, transportation control, disaster warning, and so on [1], [2]. It is challenging to provide accurate predictions since rainfall distribution is spatiotemporal specific and multiple meteorological factors are involved.

The paradigm of numerical weather prediction (NWP) [3], [4] has dominated precipitation forecasting for a few decades. NWP method is the one that is constructed on physical models for predicting the weather with current meteorological elements [5], which could make an hourly prediction with reasonable accuracy [6]. Still, it could be computationally expensive and be constrained in short-term forecasts due to the high complexities of the mathematical models [7]. Beyond that, extrapolation-based methods [8], which are faster than the NWP approach, are often adopted for precipitation nowcasting systems. Given the intensity distribution and movement tendency of the radar echo maps collected by the weather radar, radar echo extrapolation predicts a certain value in a linear or nonlinear way [9]. However, these methods are indirect prediction methods for the precipitation calculated by Z–R relation [9]. Limited by the inadequate description of the physical process between radar reflectivity factor and rain rate, there are prediction errors naturally in these methods.

Recently, deep learning techniques have dominated the machine learning fields, benefitting from massive data collecting and graphics processing unit development. Inspired by its tremendous advances in many traditional challenging tasks [10], [11], [12], [13], [14], [15], [16], researchers have applied deep learning to precipitation predictions. Current methods for precipitation nowcasting under deep learning frameworks are mainly developed on surface weather station data [17], [18], radar echo maps [19], [20], and satellite data [21], [22].

Considering the surface weather station data with non-Euclidean, graph convolutions have been employed to model the spatial correlation [23]. For example, Wilson et al. [24] proposed weighted graph convolutional LSTM (WGC-LSTM), in which the fully connected layers are replaced with graph convolutional layers within each LSTM cell. However, on account

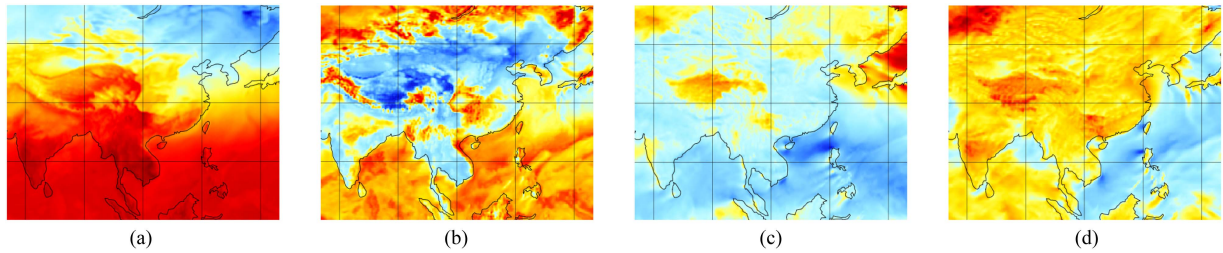


Fig. 1. Visualization of multimodal meteorological data, including temperature (K), relative humidity (%), u-component, and v-component of the wind (m/s) at 8 A.M. UTC, 1 January, 2020. (a) Temperature. (b) Relative humidity. (c) u-component of the wind. (d) v-component of the wind.

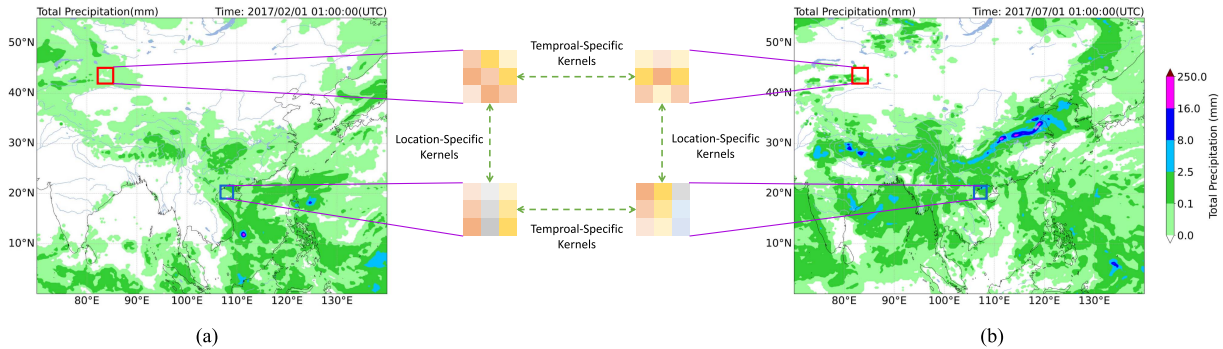


Fig. 2. Illustration of precipitation at 1 A.M. UTC, 1 February, 2016, and 1 A.M. UTC, 1 July, respectively. The deeper color means more precipitation. At the same time, the rainfall in the interior is smaller than the rainfall close to the ocean, whether in summer or winter. In addition, in the same area at different seasons, the precipitation in summer is more significant than in winter. Considering the vagaries of precipitation patterns, a spatiotemporal-specific kernel makes sense in adapting to diverse patterns with respect to different spatial positions and seasons. The figure shows that the kernels vary from position to position and season to season. (a) Hourly precipitation in winter. (b) Hourly precipitation in summer.

of the continuity of the weather system in the spatial dimension, surface weather station data subjected to discrete and sparsity are insufficient to reason weather patterns.

Technically, radar echo maps are widely employed in mainstream approaches, which convert the radar maps to predict rainfall intensity maps. Essentially, these frameworks regard precipitation nowcasting as a spatiotemporal sequence forecasting problem. For instance, Shi et al. [25] proposed convolutional LSTM (ConvLSTM) for predicting radar maps, in which fully connected structures are replaced with convolutional structures. Although approaches based on radar echo maps outperform traditional methods [26], they could not predict precipitation in an end-to-end way. Furthermore, radar echo maps are far from completely depicting the weather system whose characteristics are affected by the interaction of multiple factors.

Another source of meteorological data is geostationary satellites, which provide visible and infrared spectrum images of clouds and precipitation. In [27], Lebedev et al. used a variant of UNet on satellite imagery, including four visible and eight infrared channels, to produce nowcasting. Nevertheless, due to the orbit of satellites being higher than the troposphere, where most weather phenomenon occurs, it is hard for satellites to sense the meteorological elements.

Unlike existing deep-learning-based works, this article proposes a novelty and effective scheme for precipitation nowcasting. Specifically, as illustrated in Fig. 1, multimodal meteorological data, including temperature, relative humidity, and wind velocity along the longitude and the latitude, are used in

this article. Although deep learning could automatically learn discriminative features from a large amount of meteorological data without knowledge of meteorology, it is still challenging to employ existing studies to make predictions. For clarity, we explain the peculiarity of meteorologic data, such as spatiotemporal variance in the meteorological system, the heterogeneity of multiple modalities, and the long-tailed distribution of precipitation as follows.

Spatiotemporal Variance: Convolution neural networks are inherently subject to spatial-agnostic filters due to the local invariance in images [10]. Nevertheless, this hypothesis is not held in the meteorological system. On the one hand, precipitation does not have a uniform distribution in the temporal and spatial dimensions. For example, the precipitation in seaside areas could simultaneously be more prominent than in interior regions, as illustrated in Fig. 2. In the same region, winter precipitation is usually smaller than summer precipitation. On the other hand, the mechanism of precipitation in different latitudes and longitudes is different. For example, Wegener–Bergeron–Findeisen process frequently occurs in middle and high latitudes, which refers to the rapid growth of ice crystals in mixed-phase clouds. At the same time, the precipitation in the tropical region is mainly produced by the collision process, also called the warm rain process. Hence, the spatial-agnostic kernels are not exactly applicable to meteorological data, which could be developed nonidentically across the whole space.

Multimodal Data: In meteorology, precipitation nowcasting involves multiple weather parameters, such as surface weather

station data and wind profiler data. In this article, we consider integrating multimodal, including temperature, relative humidity, and wind velocity along the longitude and the latitude. Capturing correspondences between modalities makes it possible to gain a more thorough understanding of precipitation nowcasting. Nevertheless, due to the heterogeneity of multimodal [28], it is challenging to construct complementary representations from multiple modalities.

Long-tailed Distribution: Another significant challenge is that precipitation follows long-tail distribution: heavy rain and rainstorm infrequently occur, whereas rainless is an overwhelming majority. The long-tailed distributions [29] makes many standard approaches unlikely to fit these distributions correctly and resulting in a significant drop in less represented classes.

To solve these problems, we propose a precipitation nowcasting model named *SpatioTemporal Inference Network* (STIN). Beyond existing indirect methods, our method aims at learning a more efficient representation of the underlying dynamical and physical equations between precipitation and other multimodal meteorological elements with an end-to-end mechanism. Specifically, a spatiotemporal-aware convolutional layer (STACConv) is first designed to extract features across different regions and seasons. In STACConv, spatiotemporal-specific kernels are generated conditioned on the incoming spatiotemporal information. Based on the STACConv, we build the spatiotemporal-aware convolutional neural network (STACNN) for capturing information from multimodal meteorological data. STACNN comprises two subnetworks, the generation subnetworks and the stem-subnetworks. The generation subnetworks produce parameters spatiotemporal-specifically, and the stem-subnetworks apply the generated kernels to capture the meaningful representations of the multimodal meteorological data. Furthermore, a parameter-free multimodal fusion operation is adopted, strengthening the multimodal feature interactions across channels. Then, a ConvLSTM layer is employed to model the temporal information. Finally, a linear classifier with five intensities is adopted, and a dice loss is utilized to evaluate the long-tail distribution.

The main contributions of this article are highlighted as follows.

- 1) To predict precipitation intensity from multimodal meteorological elements, we devised a novelty and effective framework, STIN. STIN performs multimodal meteorological elements modeling dynamically and enables the forecast.
- 2) A spatiotemporal-aware convolutional layer (STACConv) is presented, which could dynamically allocate the weights over different periods and positions, leading to prioritizing the most informative meteorological modal in the spatiotemporal dimension.
- 3) An STACNN is proposed to extract multimodal spatial dependencies. An STACNN could fuse multimodal features without introducing significant extra parameters at multilayer practicality and efficiency.

II. DATASETS

1) *ERA5 Dataset:* Scratched from the public website, the ERA5 dataset is a comprehensive reanalysis of global climate

and weather for the past four decades, from 1979 to the present. It provides hourly estimates for a significant number of meteorological data, which comprise single level parameters and pressure level parameters with 37 pressure covered from 1000 hPa to 1 hPa. Both of them have been gridded to a regular 0.25° latitude–longitude grid. Since the raw dataset is enormous, we select East Asia as the study area. Such cropped region covers from 140° east to 70° west, from 55° north to the equator, with 221×281 resolutions. Also, the regional dataset comprises upper air field quantities (cropped from pressure level parameters) and land surface data (cropped from single level parameters). In the upper air field quantities, four variables of temperature, relative humidity, u-component, and v-component of the wind velocity are selected in three vertical levels: 500, 850, and 1000 hPa. The land surface data are the precipitation between adjacent observations. It should be noticed that the values at the grid denote the average physical quantities of a local region.

2) *WeatherBench Dataset:* Regridding the ERA5 data to lower resolution, WeatherBench is presented as a benchmark of global meteorological data for data-driven weather forecasting [30]. Concretely, they are the resolution of 5.625° , 2.8125° and 1.40625° at 13 vertical levels. Like the ERA5 dataset, temperature, relative humidity, u-component, and v-component of wind on the WeatherBench dataset are applied to predict precipitation. Three vertical levels are considered for vertical resolution: 500 hPa is approximately at 5.5 km height, 850 hPa is about 1.5 km height, and 1000 hPa is around sea level. Moreover, we choose the data at 1.40625° for horizontal resolution, with 128×256 resolution.

3) *RainBench Dataset:* The RainBench dataset, introduced by Witt et al. [31], serves as a comprehensive multimodal benchmark for precipitation forecasting. It comprises three categories of data: simulated satellite data, numerical reanalysis data, and global precipitation estimates. To facilitate efficient experimentation, all datasets were converted to resolutions of either 1.40625° or 5.625° from their original resolutions. In this study, we utilized the numerical reanalysis data as input and employed the global precipitation estimates as the ground truth, both at a resolution of 1.40625° .

III. PRELIMINARIES

A. Dynamic Convolution

In the past years, modern neural networks have witnessed a tremendous surge in computer vision. Considering the inherent translation-invariant of vision datasets, the core assumption of convolutional layers is that kernels should be shared for all spatial locations. Given an input \mathbf{X} and filter \mathbf{W} , the output $\mathbf{Y}_{i,j}$ at position (i, j) of standard convolutional layers is

$$\mathbf{Y}_{i,j} = (\mathbf{W} * \mathbf{X})_{i,j} \quad (1)$$

where $*$ denotes the convolution operator, and the filter \mathbf{W} is shared across spatial locations. Nevertheless, as illustrated in Fig. 2, not all data are subject to translation invariance, especially for meteorological data.

To broaden the application of standard convolutional layers, researchers emerged increasing attention to dynamic convolution. Dynamic convolution filters are generated dynamically

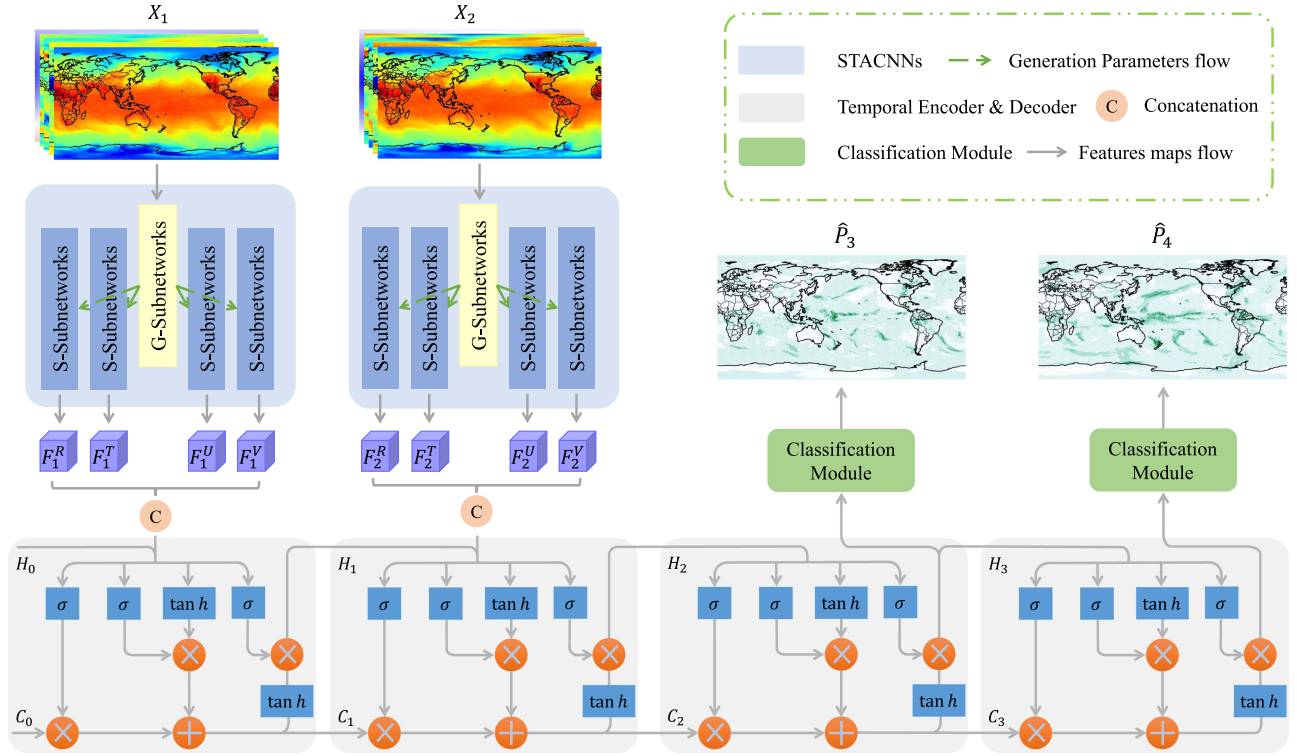


Fig. 3. Whole structure of STIN of our article. First, meteorological data at each observation are embedded by the STACNN. Then, based on the ConvLSTM, the encoder–decoder framework is employed to capture spatiotemporal dependencies and generate features in the following observations. Finally, the classification module makes predictions of rainfall intensity. “S-subnetworks” means stem subnetworks, and “G-subnetworks” means generation subnetworks.

conditioned on the input feature maps as opposed to static counterparts. Jia et al. [32] proposed dynamic filter networks in which a dynamic filter-generating network is devised to produce kernels conditioned on the input. In dynamic filter networks, the kernel is spatial-specific: for each location (i, j) of the input \mathbf{X} , the generated local kernel \mathbf{W}_θ is employed to capture features as follows:

$$\mathbf{Y}_{i,j} = (\mathbf{W}_\theta * \mathbf{X})_{i,j} = (\mathcal{G}_\theta(x) * \mathbf{X})_{i,j} \quad (2)$$

where θ is the parameters of filter-generating network \mathcal{G} . With the increase of complexity, attributing to its location-specific, dynamic convolution has more representation power and significantly boosted performance. DynamoNet [33] applied dynamic filters to capture video-specific representation for future frame prediction. Mildenhall et al. [34] proposed kernel prediction networks to denoise frames with spatial-specific kernels. Ma et al. [35] employed grouped fully connected layers to predict convolutional weight for image recognition. Despite the fact that dynamic convolution enables accuracy improvements, it increases the number of network parameters and computational complexity. Several studies have devised sophisticated dynamic kernel generation mechanisms to handle this limitation. For example, Li et al. [36] replaced dynamic convolution with the dynamic convolution decomposition method, which brings significantly fewer parameters without a performance reduction. The involution [37] leverages shared kernels along the channel dimension to alleviate the redundancy of parameters. In decoupled dynamic filter networks [38], dynamic filters are decoupled

along the spatial dimension and channel dimension to reduce the number of parameters.

B. Batch Normalization

Batch normalization (BN) [39] has been proven the core ingredient of modern neural networks, which is devised to eliminate internal covariate shifts. A BN layer first normalizes the features over a mini-batch independently and then transforms the normalized to other scales ensuring the normalizing would not constrain the representative of features. Formally, the BN layer can be performed as follows:

$$\mathbf{X}' = \gamma \frac{\mathbf{X} - \mu}{\sqrt{\delta^2 + \varepsilon}} + \beta \quad (3)$$

where \mathbf{X} and \mathbf{X}' are the input and output features of the BN layer, μ and δ are the mean and standard deviation values of input features across all locations for the current mini-batch, γ and β are the learnable scaling factor and shift factor, and ε is a small constant to avoid divisions by zero.

C. ConvLSTM

The ConvLSTM layer comprises memory cell state \mathbf{C} , hidden state \mathbf{H} , and three gates of the input gate, forget gate, and output gate. Along the temporal dimension, the forget gate \mathbf{f} is first activated to decide which memory cell state \mathbf{C} could be “forgotten.” Then, the new information would be accumulated in the memory cell state by input gate \mathbf{i} . Last, the output gate \mathbf{o} would control what information will be propagated to the next

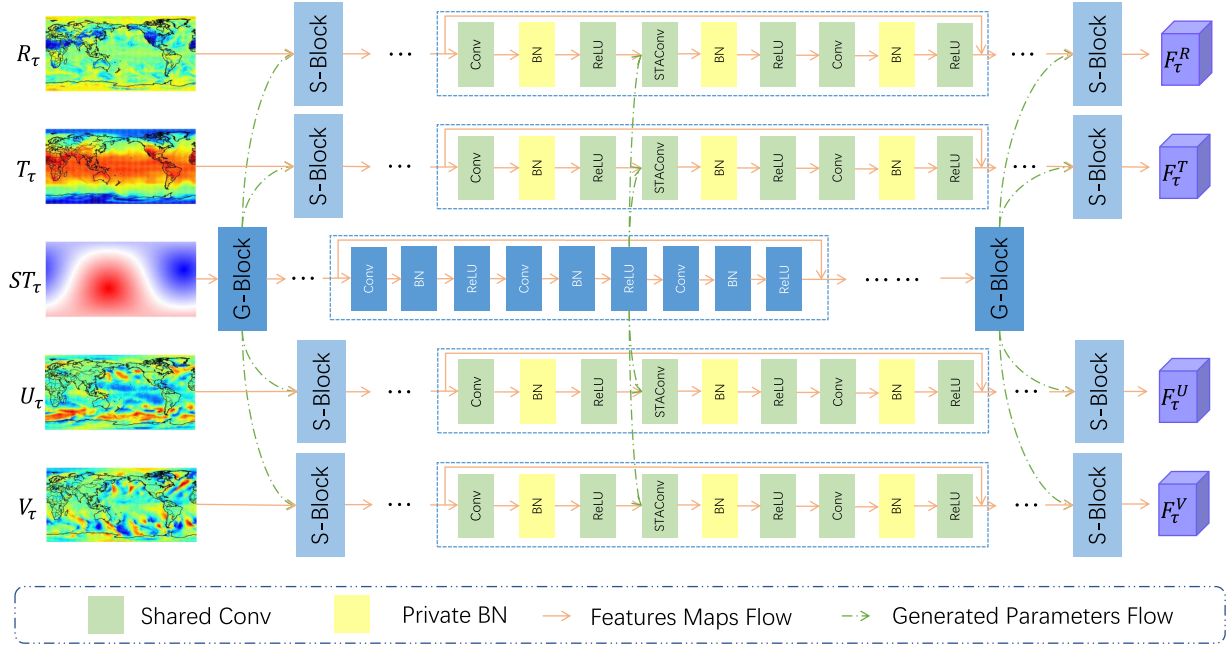


Fig. 4. Schematic illustration of STACNN. The STACNN are based on the residual block and consist of four multimodal branches for relative humidity, temperature, u-component and v-component of the wind velocity, and a spatiotemporal branch for solar elevation angle. “G-block” means generation block for embedding spatiotemporal information. “S-block” means stem block for capturing spatial features given spatiotemporal information. The convolutional layer and STACConv layer are shared between different modalities, but BN is private.

moment. Formally, given the input \mathbf{F}_τ , the ConvLSTM can be formulated as the following equations:

$$\begin{aligned}
 \mathbf{i}_\tau &= \sigma(\mathbf{W}_{xi} * \mathbf{F}_\tau + \mathbf{W}_{hi} * \mathbf{H}_{\tau-1} + \mathbf{b}_i) \\
 \mathbf{f}_\tau &= \sigma(\mathbf{W}_{xf} * \mathbf{F}_\tau + \mathbf{W}_{hf} * \mathbf{H}_{\tau-1} + \mathbf{b}_f) \\
 \mathbf{C}_\tau &= \mathbf{f}_\tau \circ \mathbf{C}_{\tau-1} + \mathbf{i}_\tau \circ \tanh(\mathbf{W}_{xc} * \mathbf{F}_\tau + \mathbf{W}_{hc} * \mathbf{H}_{\tau-1} + \mathbf{b}_c) \\
 \mathbf{o}_\tau &= \sigma(\mathbf{W}_{xo} * \mathbf{F}_\tau + \mathbf{W}_{ho} * \mathbf{H}_{\tau-1} + \mathbf{b}_o) \\
 \mathbf{H}_\tau &= \mathbf{o}_\tau \circ \tanh(\mathbf{C}_\tau)
 \end{aligned} \quad (4)$$

where “*” stands for the convolution operator and “o” denotes the Hadamard product; \mathbf{W} and \mathbf{b} is the parameters of each gate; σ is the sigmoid function.

IV. METHOD

A. Problem Formulation

Suppose there are s historical observations at the given timestamp τ . We use \mathbf{R} , \mathbf{T} , \mathbf{U} , and \mathbf{V} to denote the inputs of the relative humidity, temperature, u-component, and v-component of the wind velocity successively. Furthermore, \mathbf{R}_τ , \mathbf{T}_τ , \mathbf{U}_τ , $\mathbf{V}_\tau \in \mathbb{R}^{l \times m \times n}$ represent the value at τ th observation, where l enumerates pressure in hecto-Pascals as a vertical coordinate and a spatial region is represented by an $m \times n$ grid. To simplify the notation, the observation at given timestamp τ can be represented by a tensor $\mathbf{X}_\tau \in \mathbb{R}^{4 \times l \times m \times n}$ that consists of \mathbf{R}_τ , \mathbf{T}_τ , \mathbf{U}_τ , and \mathbf{V}_τ . Given the previous s observations (including the current one), the goal of precipitation nowcasting is to predict the most likely precipitation sequence of length j in the future

$$\hat{\mathbf{P}}_{\tau+1}, \hat{\mathbf{P}}_{\tau+2}, \dots, \hat{\mathbf{P}}_{\tau+j} = \mathcal{F}(\mathbf{X}_{\tau-s+1}, \mathbf{X}_{\tau-s+2}, \dots, \mathbf{X}_\tau) \quad (5)$$

where $\mathbf{P}_{\tau+1} \in \mathbb{N}^{m \times n}$ denotes precipitation between τ th observation and $\tau + 1$ th observation in a certain area.

B. Overview

To address the precipitation nowcasting with multimodal meteorological data, we propose the STIN. STIN includes an STACNN, an encoder–decoder framework, and a classification module, as shown in Fig. 3. Specifically, considering the spatiotemporal variance of meteorological data, the STACNN is first applied to capture the spatial features of the raw multimodal meteorological data $\mathbf{X} \in \mathbb{R}^{4 \times l \times m \times n}$ from $\tau - s + 1$ th observation to τ th observation. The overall structure of the STACNN is illustrated in Fig. 4, which consists of two subnetworks, a generation subnetworks, and the stem subnetworks. The generation subnetworks generate filters to capture the spatiotemporal variance representations conditioned on spatiotemporal information. Then, the stem subnetworks allocate the generated filters over each spatiotemporal location of the input, helping the spatiotemporal-aware convolutional layer construct the dynamic representations. Meanwhile, a multimodal fusion strategy is exploited to fuse multimodal features at multiple stages of the network. Formally, let $\mathbf{F} \in \mathbb{R}^{c_f \times m_f \times n_f}$ denote the multimodal spatial representations, where c_f is the number of channels, and m_f and n_f represent the height and weight of features, respectively. Then, given the spatial representations $\mathbf{F}_{\tau-s+1}, \mathbf{F}_{\tau-s+2}, \dots, \mathbf{F}_\tau$, the spatiotemporal encoder network is employed to model the spatiotemporal dynamics. Along the temporal dimension, the decoder network exploits the spatiotemporal representations to generate the predicting features $\mathbf{H}_\tau, \mathbf{H}_{\tau+1}, \dots, \mathbf{H}_{\tau+j}$ in the next j observations. Finally, the

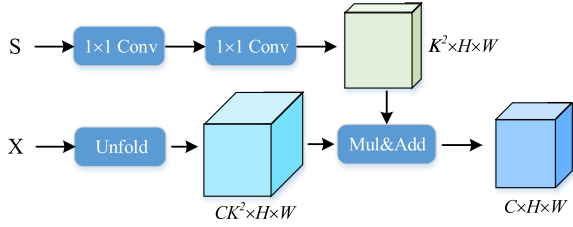


Fig. 5. Schema of the STACConv. With the spatiotemporal information as auxiliary input, we employ the nonlinear transformation f described in (8) to generate kernels dynamically. “S” stands for the spatiotemporal information and “X” denotes the input features.

classifier is applied to predict rainfall intensity by employing the dice loss [40].

C. SpatioTemporal-Aware Convolutional Neural Network

This section describes our STACNN, which includes two subnetworks, a generation subnetworks that produce kernels with spatiotemporal information and a stem subnetworks that applies the generated kernels to the input. The generation subnetworks take the spatiotemporal information $\mathbf{S} \in \mathbb{R}^{c_s \times m_s \times n_s}$ as input, where c_s , m_s , and n_s are the number of channels, height, and width of spatiotemporal information \mathbf{S} , respectively. It outputs spatiotemporal variance kernels \mathbf{W}_θ parameterized by parameters θ . The generated kernels are employed to capture features.

1) *SpatioTemporal-Aware Convolutional Layer*: The standard convolution operation allocates the same weights over different positions. The shared kernels reduce parameter load and capture translationally invariant features, which would be helpful in some applications, such as computer vision. Let $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ stand for the input, where C denotes the input channels and H and W represents the height and weight, respectively, and $\mathbf{W} \in \mathbb{R}^{C \times K \times K}$ with the fixed size of $K \times K$ stands for the kernel, the output $\mathbf{Y}_{k,i,j}$ at the k th channel on location (i, j) is

$$\mathbf{Y}_{k,i,j} = \sum_{c=0}^C \sum_{p=0}^K \sum_{q=0}^K \mathbf{W}_{c,p,q} \cdot \mathbf{X}_{c,i-[K/2]+p,j-[K/2]+q}. \quad (6)$$

It allows convolution to filter data with translation invariance, such as images. Nevertheless, due to the spatiotemporal variance of meteorological data, dynamic convolution is more suitable for the precipitation nowcasting problem. Thus, spatiotemporal-aware convolutional (STACConv) layer is developed to eliminate the invariance. Technically, we generalize the kernels in the classical convolution and model the spatiotemporal information at the vertex into the generalized filters as illustrated in Fig. 5. Specifically, given spatiotemporal information \mathbf{S} aligned to input \mathbf{X} and a kernel generation function \mathcal{G} , the STA convolution can be defined as

$$\mathbf{Y}_{k,i,j} = \sum_{c=0}^C \sum_{p=0}^K \sum_{q=0}^K \mathcal{G}(\mathbf{S}_{i,j})_{[cg/C],p,q} \cdot \mathbf{X}_{c,i-[K/2]+p,j-[K/2]+q} \quad (7)$$

where g is a hyperparameter to reduce the interchannel redundancy inspired by RedNet [37].

Unlike RedNet and standard convolution filters, the kernels of STACConv rely on spatiotemporal information, which can be encoded via alternative data, such as a concatenation of longitude, latitude, and time. Previous meteorological-related work has also recognized the importance of spatiotemporal information. For example, Bai et al. [41] proposed a perceptron to extract the spatiotemporal information from season, month, date stamp, geographic longitude, and latitude. Zhao et al. [42] introduced geo-queries to model similar semantic ingredients by embedding different labels with spatiotemporal information. In this work, spatiotemporal information S is generated, conditioned on a solar elevation angle α_s , which is the angle between the sun’s rays and the horizontal plane. Specifically, benefits can be listed as follows. On the one hand, the solar elevation angle is a hand-crafted feature obtained through a nonlinear mapping of temporal and spatial information, calculated as follows:

$$\cos \alpha_s = \sin \Phi \sin \delta + \cos \Phi \cos \delta \cos h_s \quad (8)$$

where h_s is the local solar time, Φ is the local latitude, and δ is the current declination of the sun. On the other hand, the solar radiation is the primary driving energy for the movement of the Earth’s atmosphere, and the solar elevation angle can reflect solar radiation intensity.

To build the entire STACNN, the encoder of the stem subnetworks and generation subnetworks are first described. Then, the decoder of stem subnetworks is depicted to recover the resolution of features. For the encoder of the stem subnetworks, we follow the same spirit as ResNet, which stacks residual blocks for capturing different scale features. We remain all the 1×1 convolution in bottleneck blocks of ResNet unchanged but replace the 3×3 convolution with 7×7 STA convolution. By the way, the spatiotemporal information \mathbf{S} should be comfortably aligned to the input at different stages to generate kernels.

To produce spatial alignment spatiotemporal information from the solar elevation angle, a ResNet50 network is employed within the generation subnetworks. We take the output of 3×3 convolution in ResNet50 bottleneck blocks as spatiotemporal information \mathbf{S} for which spatial and channel information interleaves simultaneously, ensuring the representation capability. Then, to span each kernel from spatiotemporal information \mathbf{S} , a simple nonlinear transformation $\mathcal{G} : \mathbb{R}^C \mapsto \mathbb{R}^{K \times K \times G}$ is devised as follows:

$$\mathcal{G}(\mathbf{S}_{i,j}) = \mathbf{W}_2(\text{ReLU}(\text{BN}(\mathbf{W}_1 \mathbf{S}_{i,j}))) \quad (9)$$

where $\mathbf{W}_1 \in \mathbb{R}^{\frac{C}{\gamma} \times C}$ and $\mathbf{W}_2 \in \mathbb{R}^{K \times K \times G \times \frac{C}{\gamma}}$ form a two-layer bottleneck block with a reduction ratio γ to reduce dimensions. In a word, the spatiotemporal information generated by the channel–spatial interactions is applied to implicitly encode precipitation patterns via channel information exchange (realized by \mathcal{G}). After that, the spatiotemporal information in a neighborhood is aggregated dynamically. Different from us, the DY-CNNs [43] enhances model capacity by aggregating multiple convolutional kernels through attention mechanisms. However, their dynamically generated convolution still conforms to the

spatial-agnostic, which is not conducive to meteorological data modeling.

In the decoder of the stem subnetworks, an atrous spatial pyramid pooling (ASPP) [44] module, which is made up of five parallel pathways with different receptive field, is first applied to extract and fuse multiscale features. Then, to recover the resolution of feature maps, we mirror the decoder in DeepLabv3plus [44]. Concretely, the output of ASPP is upsampled by a factor of four to align with the low-level features. Then, low-level features are fed into a 1×1 convolution to reduce the number of channels. Afterward, we apply a 3×3 convolution to refine the concatenation of the high-level and low-level features. We can obtain spatial features that could be utilized as input to the encoder–decoder framework.

D. Multimodal Fusion

As mentioned before, the precipitation nowcasting problem involves multiple weather elements, and four modalities are chosen in this work. However, given the heterogeneity of the multimodal data, some unique challenges are brought for machine learning researchers, such as aligning features from different sources [28].

Extensive research has been conducted on multimodal fusion. For instance, Zhuang et al. [45] introduced additional information to enhance spatial fusion. Guo et al. [46] proposed a nonlinear fusion strategy for spatial fusion in the multigradient domains. TransFuser [47] developed a multimodal fusion transformer. Nagrani et al. [48] utilized cross-attention to fuse multimodal features. The CSMFormer [49] is designed to fuse hyperspectral and multispectral images by combining the long-range dependencies and the local information. Gao et al. [50] proposed DFINet to extract self-correlation and cross correlation from multimodal feature pairs. The ACL-CNN [51] integrate the adversarial complementary learning strategy into the CNN to extract the complementary information of the multimodal data. In the MDA-NET [52], a mutual-aid classifier is used to aggregate all the discriminative features for different modalities. Zhang et al. [53] developed SOT-Net for strengthening the semantic relatedness of multimodal data.

However, these methods are not well suited to our specific task. For fusing multimodal features as in previous work, individual STACNN could be applied to different modalities. Except that individual networks bring heavy parameters, it also hinders the implicit fusing of multimodal. In multiple tasks or domains, sharing convolution layer parameters with independent BNs has been leveraged to be effective for model adaption [54], [55]. Inspired by this, we boost this idea for dynamic convolution. The generation subnetworks and stem subnetworks are shared for all modalities except the normalization and transformation of multimodal acts separately. On the one hand, we can vastly reduce model parameters for learning features from multimodal meteorological data. On the other hand, features from different modalities could interact implicitly with each other through the shared convolutional layer.

To fuse the multimodal information at multiple layers, we adapt the channel shuffle operation [56]. Concretely, we split

the features into four groups by channels, given the four feature maps \mathbf{X}^R , \mathbf{X}^T , \mathbf{X}^U , and \mathbf{X}^V . Then, four feature maps are exchanged with each other as follows:

$$\begin{aligned}\mathbf{X}_{\text{new}}^R &= \text{concat}(\mathbf{X}_1^R, \mathbf{X}_1^T, \mathbf{X}_1^U, \mathbf{X}_1^V) \\ \mathbf{X}_{\text{new}}^T &= \text{concat}(\mathbf{X}_2^R, \mathbf{X}_2^T, \mathbf{X}_2^U, \mathbf{X}_2^V) \\ \mathbf{X}_{\text{new}}^U &= \text{concat}(\mathbf{X}_3^R, \mathbf{X}_3^T, \mathbf{X}_3^U, \mathbf{X}_3^V) \\ \mathbf{X}_{\text{new}}^V &= \text{concat}(\mathbf{X}_4^R, \mathbf{X}_4^T, \mathbf{X}_4^U, \mathbf{X}_4^V)\end{aligned}\quad (10)$$

where \mathbf{X}_1^R , \mathbf{X}_2^R , \mathbf{X}_3^R , and \mathbf{X}_4^R are features of the first, second, third, and fourth groups, respectively.

E. Encoder–Decoder Framework

Given the spatial features $\mathbf{F}_{\tau-s+1}, \mathbf{F}_{\tau-s+2}, \dots, \mathbf{F}_{\tau}$ from $\tau - s + 1$ th observation to τ th observation, we further explore the temporal dependencies by RNN-based models. The RNN-based models form an encoder–decoder structure in which the precipitation in the following few observations could be predicted successively. Here, the ConvLSTM layer is employed as an ingredient of the encoder–decoder structure. Specifically, at the τ th observation, the ConvLSTM layer take multimodal meteorological features \mathbf{F}_{τ} , previous hidden state $\mathbf{H}_{\tau-1}$, and previous cell state $\mathbf{C}_{\tau-1}$ as input. In the following j observations, the ConvLSTM layer generates predicting features conditioned on the previous hidden state and cell state.

F. Supervised Learning

At last, a classification module is adopted to predict rainfall intensity given $\mathbf{H}_{\tau}, \mathbf{H}_{\tau+1}, \dots, \mathbf{H}_{\tau+j}$. With a stacked convolutional layer and activation functions, the classification module makes predictions $\hat{\mathbf{P}}_{\tau+1}, \hat{\mathbf{P}}_{\tau+2}, \dots, \hat{\mathbf{P}}_{\tau+j}$. To align the predictions with input, the original $\hat{\mathbf{P}}_{\tau+1}, \hat{\mathbf{P}}_{\tau+2}, \dots, \hat{\mathbf{P}}_{\tau+j}$ are upsampled by a factor of four. To relieve the imposed by long-tailed distribution, we adapt the dice loss [40] to provide supervision information. The dice loss can alleviate biases imposed by head classes. Given predicted categorical probability $p_{i,j}(\hat{\mathbf{Y}}_c|\mathbf{X})$ at pixel i, j w.r.t. its categorical i . Then, the dice loss at pixel i, j is written as follows:

$$\text{Loss} = \frac{\sum_{c=1}^C p_{i,j}(\hat{\mathbf{Y}}_c|\mathbf{X}) \mathbf{Y}_c}{\sum_{c=1}^C p_{i,j}^2(\hat{\mathbf{Y}}_c|\mathbf{X}) + \sum_{c=1}^C \mathbf{Y}_c}.\quad (11)$$

G. Theoretical Comparison to NowcastNet

The NowcastNet [57] and our method are both designed to predict precipitation in the short term. However, the two methods adopt distinct approaches in modeling precipitation prediction, leading to a customized framework. In NowcastNet, precipitation forecasting is viewed as video generation that predicts future radar ds given past radar fields. A deep generative model is introduced to generate predictions from the physics-informed evolutions and latent Gaussian vector. At the same time, our method addresses the task of precipitation forecasting as a multimodal spatiotemporal prediction problem that takes multiple meteorological data as its input and directly predicts precipitation in the short term. The STIN is introduced

to learn representative spatiotemporal-specific correlations between precipitation and multiple meteorological data.

V. EXPERIMENTS

To verify the effectiveness of our model, we conducted several experiments, which can be listed as follows: First, we estimate our model's performance on precipitation nowcasting. For both datasets, STIN and comparison models are trained from scratch to predict hourly precipitation in the next six hours, which receives hourly multimodal meteorological data from the past six hours as input. Then, we evaluate the effectiveness of each component in our model, including the STACNN, the multimodal fusion, and the encoder–decoder framework. Moreover, extensive ablation experiments are performed to explore the hyperparameters, such as the number of training epochs, the scale of training data, and the learning rate.

A. Implementation

1) *Experiment Settings*: For the ERA5 and WeatherBench datasets, STIN is trained for 20 epochs with the standard Adam optimizer. The initial learning rate is set to 5×10^{-4} , and the learning rate is decayed as exponential with the gamma of 0.9. The batch size is set to 32, and the training is conducted on eight NVIDIA RTX GPUs. The ERA5 regional dataset contains hourly meteorological data from 2016 to 2020, and the WeatherBench dataset used in this article is a subset of five years from 2014 to 2018. For both of them, the meteorological data of the first three years are used as the training set, and that of the fourth year and fifth year are adopted as the validation and testing dataset, respectively. For the ERA5 dataset, a 192×256 patch is cropped from the regional data. After that, the upper air field quantities are normalized. For the WeatherBench dataset, normalization is adopted on the modalities.

2) *Evaluation Metrics*: This article predicts hourly precipitation intensity in the next six hours, given the historical meteorological elements of the past six hours. To evaluate the performance of the proposed method quantitatively, multiple criteria are used, including the threat score (TS), the false alarm ratio (FAR) as well as the missing alarm rate (MAR), and the intersection over union (IoU). The TS and IoU metrics assess the overlap between the predicted and actual precipitation areas. The FAR quantifies the proportion of forecasted precipitation areas where no actual precipitation occurs, whereas the MAR signifies the proportion of missed precipitation areas within the genuine precipitation region. Among them, TS, FAR, and MAR are primarily used in the meteorological field, and IoU is extensively employed in machine learning. The threshold of precipitation intensity is illustrated in Fig. 6, drawing from the research by Jebson [64]. IoU is defined as follows:

$$\text{IoU} = \frac{\hat{P} \cap P}{\hat{P} \cup P} \quad (12)$$

where \hat{P} is the predictions and P is the ground truth, and \cap and \cup means intersection and union operations, respectively. Unlike the IoU, which evaluates the precision of predictions, the meteorological metrics allow the model to make larger predictions of precipitation intensity. In meteorology, the positive sample is

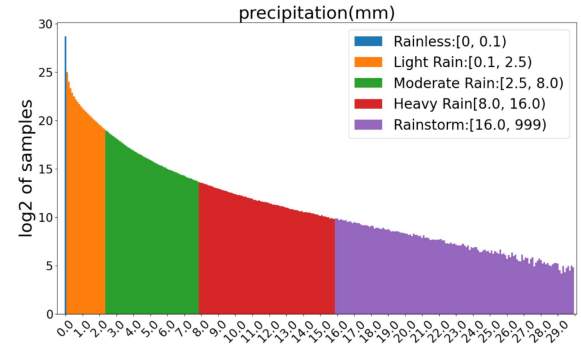


Fig. 6. Distribution of precipitation intensity in an hour on the ERA5 dataset. We classify precipitation intensity as the threshold shown in the upper right corner.

defined when the value at the grid point is equal to or greater than the threshold. Contrary, when the value at the grid point is equal to or less than the threshold, the sample is defined as negative. Before giving the calculation of TS, FAR, and MAR, we first introduce hits, correctnegatives, falsealarms, and misses

$$\begin{aligned} \text{hits} &= (P \geq \text{TH}) \& (\hat{P} \geq \text{TH}) \\ \text{correctnegatives} &= (P < \text{TH}) \& (\hat{P} < \text{TH}) \\ \text{falsealarms} &= (P < \text{TH}) \& (\hat{P} \geq \text{TH}) \\ \text{misses} &= (P \geq \text{TH}) \& (\hat{P} < \text{TH}). \end{aligned} \quad (13)$$

After that, the TS, FAR, and MAR can be defined as follows:

$$\begin{aligned} \text{TS} &= \frac{\text{hits}}{\text{hits} + \text{misses} + \text{falsealarms}} \\ \text{FAR} &= \frac{\text{falsealarms}}{\text{hits} + \text{falsealarms}} \\ \text{MAR} &= \frac{\text{misses}}{\text{hits} + \text{misses}}. \end{aligned} \quad (14)$$

For TS, FAR, and MAR, it should be noted that the rainless is ignored in their calculation. This article pays more attention to the IoU and TS; other metrics are only for reference.

B. Comparisons

Several models are compared with the proposed method, including statistical methods and deep learning-based methods.

1) *Baselines. Statistical methods*: Statistical methods include climatology, weekly climatology, and persistence, all proposed in WeatherBench [30]. For climatology, the prediction is the mean of precipitation over the training dataset. Except for climatology, weekly climatology is another straightforward but practical method. Given the periodicity of meteorology, the average weekly precipitation in adjacent years is highly related. For weekly climatology, the average precipitation of each week is first computed, and then the corresponding week takes the average weekly precipitation as the prediction. Persistence is another simple method in which precipitation in the following observations is close to the past rainfall. In this article, we take

TABLE I
OVERALL PERFORMANCES OF THE PROPOSED METHOD ON ERA5 DATASET

Metric	Method	Forecast time (hours)					
		0~1	1~2	2~3	3~4	4~5	5~6
IoU↑	Pers. [30]	31.10	28.64	26.98	25.78	24.89	24.22
	Clim. [30]	13.36	13.36	13.36	13.36	13.36	13.36
	W-Clim. [30]	16.68	16.68	16.68	16.68	16.68	16.68
	ConvLSTM [25]	40.52	40.10	38.84	37.39	35.85	34.29
	ConvGRU [58]	40.01	39.71	38.83	37.75	36.56	35.31
	TrajGRU [59]	40.57	40.12	36.45	35.02	32.72	31.63
	PredRNN [60]	40.56	39.75	37.49	35.67	33.59	32.39
	MIM [61]	36.82	36.89	35.58	33.83	32.39	30.82
	CausalLSTM [62]	36.68	36.48	35.23	34.22	33.12	31.45
	PFST [63]	40.67	40.43	39.29	37.56	35.94	34.42
	Ours	43.15	42.95	41.76	40.13	38.16	36.06
TS↑	Pers. [30]	15.27	13.57	12.38	11.125	10.86	10.34
	Clim. [30]	6.36	6.36	6.36	6.36	6.36	6.36
	W-Clim. [30]	7.24	7.24	7.24	7.24	7.24	7.24
	ConvLSTM [25]	22.48	22.15	21.32	20.39	19.40	18.37
	ConvGRU [58]	21.89	21.63	20.98	20.20	19.35	18.45
	TrajGRU [59]	21.87	21.62	20.22	18.84	17.61	16.85
	PredRNN [60]	20.13	19.75	18.62	17.48	16.34	15.43
	MIM [61]	20.92	20.34	19.27	17.65	16.23	15.14
	CausalLSTM [62]	19.45	19.13	18.55	17.86	17.09	15.88
	PFST [63]	20.06	20.18	19.58	18.78	17.95	17.12
	Ours	22.89	22.83	22.06	20.96	19.77	18.85

The results reported in this Table are the mean value covering all categories where the values in bold are the best.

the average precipitation from the past six hours as the prediction of hourly precipitation in the next six hours.

Deep learning-based methods: To prove the effectiveness of our method, STIN is compared with ConvLSTM, ConvGRU, TrajGRU, PredRNN, MIM, CausalLSTM, and PFST. Although these methods are designed for sequence forecasting problems, for fair comparisons, they all take the temperature, relative humidity, and wind as input to predict the following precipitation. For the encoder–decoder structure, such as ConvLSTM, ConvGRU, and TrajGRU, we follow the network architecture proposed in TrajGRU [59] where all of them contain encoding network and forecasting both formed by stacking RNN layers. Apart from these, PredRNN, MIM, CausalLSTM, and PFST are reimplementation to adapt to our task where input and output are different.

C. Result

In this section, STIN is compared with the comparison methods on the three datasets. The results of the experiments are given in Tables I–III, and Fig. 9, which illustrate the performance of the comparison methods. Meanwhile, the visualization of prediction errors is shown in Fig. 7. Table I reports the IoU and TS of different methods on the ERA5 dataset. Compared with the statistical methods, STIN achieves a considerable margin of higher overall metrics due to the dynamic of the meteorological system. Likewise, it is seen that similar performance gains in comparison with deep learning-based methods. For the WeatherBench dataset, STIN still achieves better performance than others, as given in Table II. Nevertheless, due to the grid of WeatherBench being bigger than the regional dataset, quantities at every grid are smoother statistically, leading to a more imbalanced distribution of precipitation. The more imbalanced distribution results in a smaller margin than other methods. Different from the indirect methods such as sequence-to-sequence learning which predict

TABLE II
OVERALL PERFORMANCES OF THE PROPOSED METHOD ON THE WEATHERBENCH DATASET

Metric	Method	Forecast time (hours)					
		0~1	1~2	2~3	3~4	4~5	5~6
IoU↑	Pers. [30]	28.50	27.94	26.20	24.95	24.03	23.33
	Clim. [30]	16.10	16.10	16.10	16.10	16.10	16.10
	W-Clim. [30]	17.04	17.04	17.04	17.04	17.04	17.04
	ConvLSTM [25]	29.56	29.39	29.05	28.69	28.25	27.93
	ConvGRU [58]	27.64	27.55	27.23	26.86	26.53	26.14
	TrajGRU [59]	29.32	29.21	28.85	28.65	28.24	27.90
	PredRNN [60]	29.33	29.28	28.95	28.55	28.07	27.63
	MIM [61]	26.89	26.97	27.03	26.65	26.22	25.67
	CausalLSTM [62]	28.15	27.95	27.73	27.42	27.16	26.85
	PFST [63]	25.96	25.63	26.08	26.54	26.87	26.83
	Ours	30.37	31.23	31.14	30.83	30.30	29.51
TS↑	Pers. [30]	14.09	12.18	10.95	9.86	9.12	8.54
	Clim. [30]	5.64	5.64	5.64	5.64	5.64	5.64
	W-Clim. [30]	5.81	5.81	5.81	5.81	5.81	5.81
	ConvLSTM [25]	15.58	15.33	15.02	14.74	14.44	14.03
	ConvGRU [58]	14.95	14.86	14.43	14.25	13.93	13.62
	TrajGRU [59]	15.43	15.34	14.99	14.71	14.41	14.08
	PredRNN [60]	15.16	15.18	14.73	14.42	14.02	13.61
	MIM [61]	13.74	13.98	13.99	13.74	13.37	12.94
	CausalLSTM [62]	14.85	14.73	14.52	14.28	14.03	13.77
	PFST [63]	11.14	10.93	11.39	11.94	12.27	12.41
	Ours	15.71	15.83	15.59	15.31	14.95	14.32

The results reported in this Table are the mean value covering all categories where the values in bold are the best.

TABLE III
OVERALL PERFORMANCES OF THE PROPOSED METHOD ON RAINBENCH DATASET

Metric	Method	Forecast time (hours)					
		0~1	1~2	2~3	3~4	4~5	5~6
IoU↑	Pers. [30]	27.16	25.22	23.96	23.07	22.43	21.95
	Clim. [30]	14.17	14.17	14.17	14.17	14.17	14.17
	W-Clim. [30]	15.93	15.93	15.93	15.93	15.93	15.93
	ConvLSTM [25]	26.16	26.13	26.12	26.09	26.03	25.90
	ConvGRU [58]	25.44	25.43	25.42	25.34	25.14	24.94
	TrajGRU [59]	25.60	25.61	25.56	25.33	25.46	25.39
	PredRNN [60]	24.83	24.44	24.02	23.69	23.43	23.22
	MIM [61]	24.27	24.21	24.02	23.95	23.58	23.55
	CausalLSTM [62]	26.72	26.70	26.66	26.64	26.54	26.30
	PFST [63]	24.39	24.28	24.03	23.69	23.30	22.87
	Ours	27.66	27.48	27.35	27.13	26.89	26.76
TS↑	Pers. [30]	9.68	8.05	6.95	6.17	5.60	5.17
	Clim. [30]	2.92	2.92	2.92	2.92	2.92	2.92
	W-Clim. [30]	3.10	3.10	3.10	3.10	3.10	3.10
	ConvLSTM [25]	10.47	10.46	10.43	10.43	10.38	10.27
	ConvGRU [58]	10.70	10.69	10.68	10.62	10.47	10.33
	TrajGRU [59]	10.75	10.74	10.71	10.66	10.63	10.53
	PredRNN [60]	8.38	7.92	7.43	7.01	6.68	6.41
	MIM [61]	9.18	9.14	9.00	8.92	8.67	8.57
	CausalLSTM [62]	9.33	9.31	9.30	9.25	9.17	9.03
	PFST [63]	7.14	6.92	6.55	6.07	5.47	4.71
	Ours	11.74	11.69	11.36	11.25	11.06	10.42

The results reported in this Table are the mean value covering all categories where the values in bold are the best.

radar echo sequences with historical radar echo maps, our work employs multi-modal data including temperature, humidity, wind speed, and others to predict precipitation. This is because precipitation is a multifaceted weather phenomenon arising from the interaction of multiple meteorological elements. Thus, compared with the sequence forecasting methods, we devise STACNN exquisitely for capturing multimodal spatial features dynamically and simplifying the encoder–decoder framework.

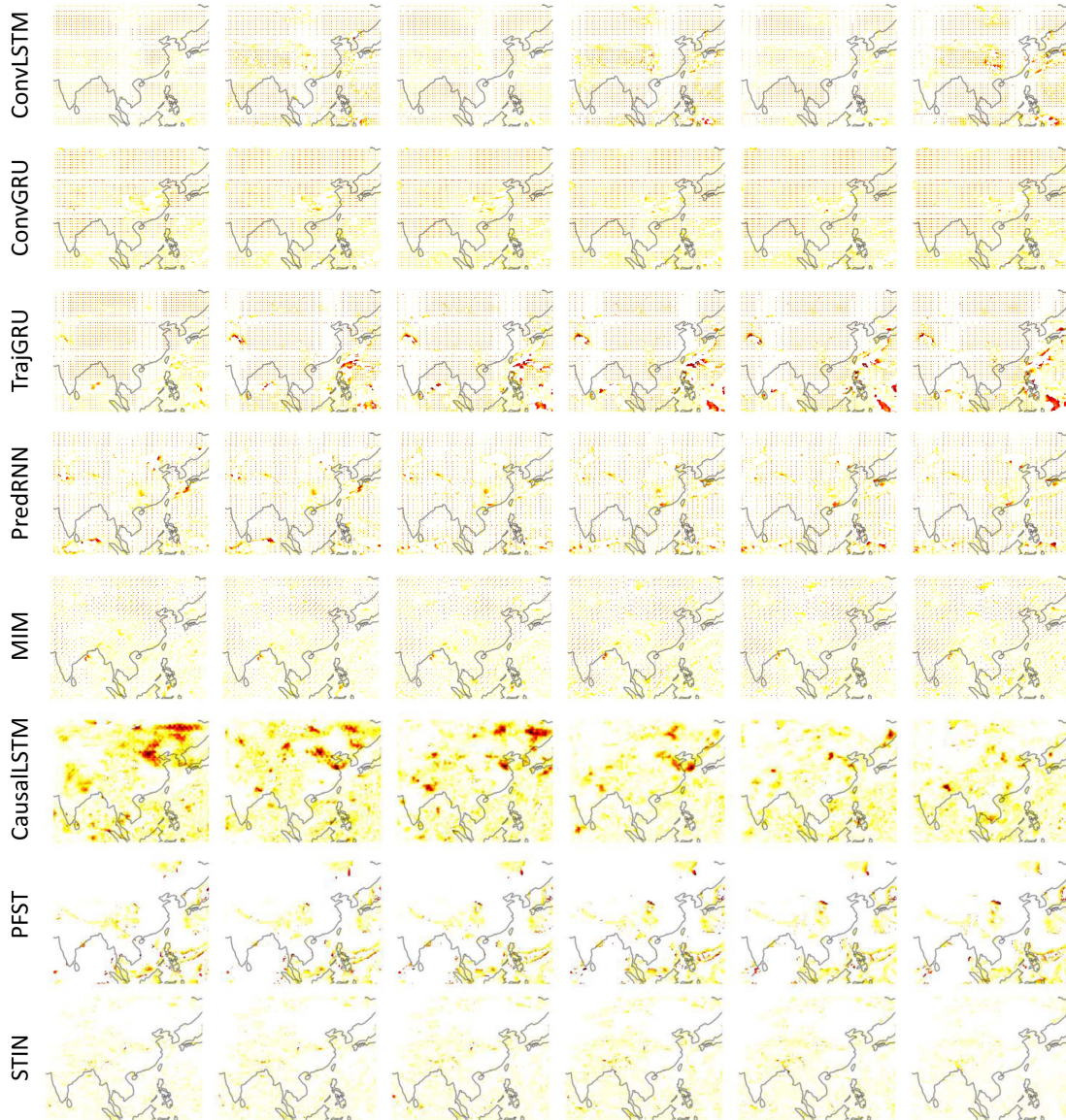


Fig. 7. Visualization of prediction errors is calculated by cross-entropy of each deep learning-based method. the deeper color means more error.

The performance gains can be briefly concluded that spatial feature extraction is more significant than modeling temporal dependencies in sequence inference problems, which could be proved by comparing STIN with ConvLSTM. Tables I and II also report the TS of different methods. The performance of our method is optimal or suboptimal.

Furthermore, Fig. 9 conveys the FAR and MAR of each method. Our model performs better than other approaches on MAR and only falls behind PFST on FAR. STIN tries to produce distinct decision boundaries for each level as far as possible, whereas PFST biases the decision boundaries toward dominated classes, resulting in fewer predictions on the tail class. On the one hand, the fewer predictions lead to a more significant amount of missing alarms and fewer false alarms of the tail class. On the other hand, fewer predictions in the tail class mean larger predictions in other classes, contributing to the larger false alarms in other classes. Thus, PFST performs better on FAR but degrades

TABLE IV
RESULTS ON THE ERA5 DATASET OF EACH PRECIPITATION NOWCASTING FAR AND MAR FOR MULTIPLE LEVELS

Metric	Time	Model	L.R.	M.R.	H.R.	RS
FAR	3	PFST	32.20	47.01	51.13	39.18
		Ours	29.46	46.19	50.76	55.91
	6	PFST	37.07	59.72	65.79	56.46
		Ours	34.20	56.03	63.06	64.62
MAR	3	PFST	28.49	69.05	75.24	84.62
		Ours	27.87	53.73	64.26	79.53
	6	PFST	31.26	68.36	81.68	94.76
		Ours	29.32	67.29	80.16	92.45

The bold values indicate the best performance.

on MAR of the rainstorm, whereas STIN achieves better performance on light, moderate, and heavy rain for both metrics, as reported in Table IV. However, in practical forecasting, MAR is

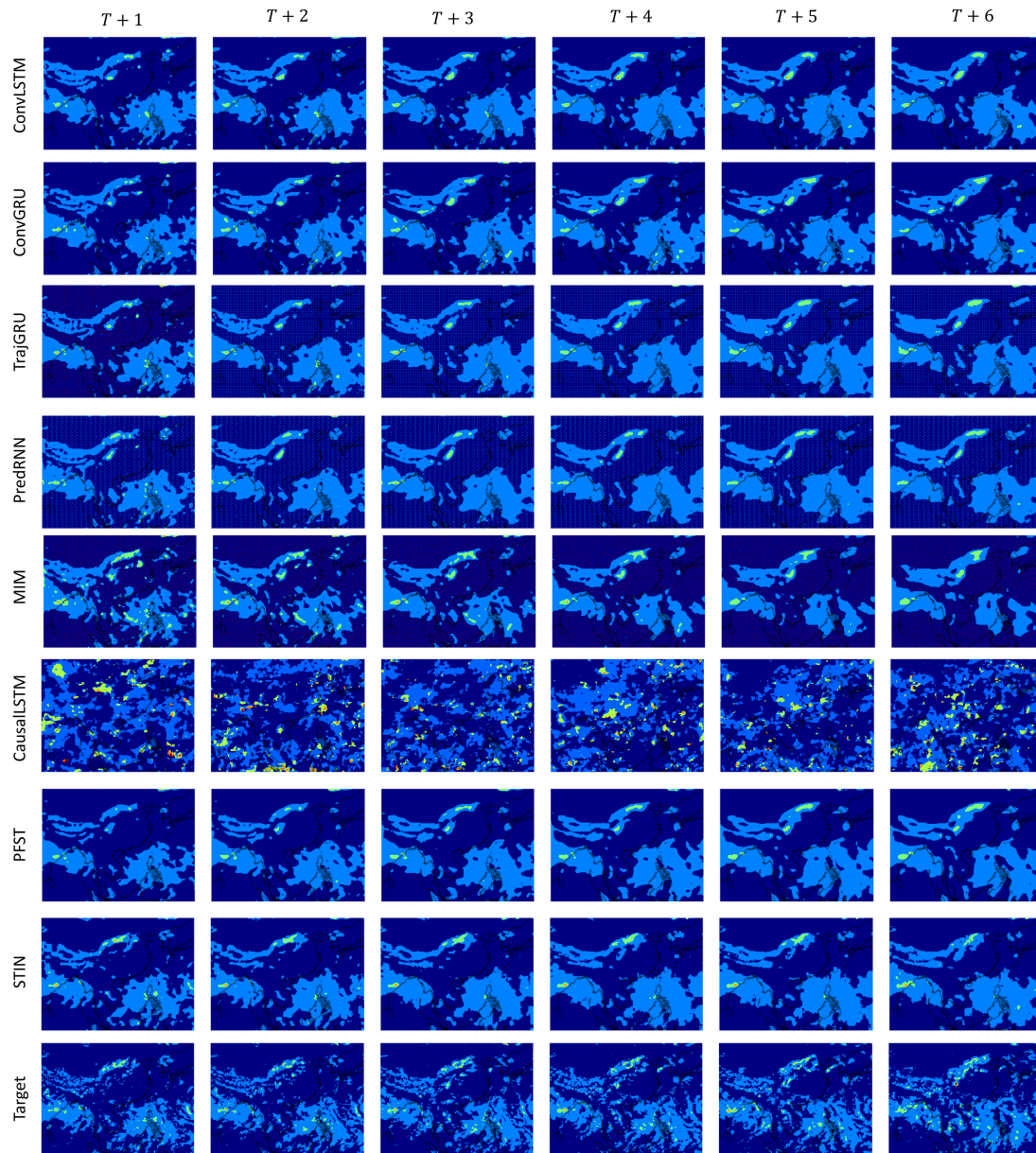


Fig. 8. Case study of performance on precipitation starting on 2019-06-17 at 23:00. Compared with other models, STIN has predicted heavy precipitation events in eastern India across several lead times.

more significant than FAR because missing alarms are more hazardous than false alarms, especially for extreme weather, such as rainstorms (i.e., tail class). The mean values of TS, IoU, MAR, and FAR overall forecast times are reported in Table III for the RainBench dataset. The results indicate that the STIN outperforms other methods across all metrics. These findings suggest that the key to achieving excellent performance lies in the elaborate structure and convolutional layer that dynamically captures multimodal meteorological features.

As shown in Fig. 7, we visualize the prediction errors calculated by cross-entropy that darker color means greater error. Our method is lighter than others, demonstrating that the STIN achieves superior performance. The errors that grow with sequence increases are consistent with objective phenomena and laws. The STIN outperforms other models on the overall

sequence. Moreover, we provide a case study for comparison with our model and other methods. Fig. 8 depicts heavy precipitation occurring in eastern India within the next five hours. The ConvLSTM, CausalLSTM, and PFST fail to predict this heavy precipitation event. The ConvGRU, TrajGRU, PredRNN, and MIM are only able to predict heavy precipitation in a few lead times. In comparison with these methods, STIN has predicted heavy precipitation events across several lead times.

Analyzing the convergence of the training process for deep models poses significant challenges due to the highly nonlinear nature of deep models and the utilization of mini-batches in each iteration. To attain a satisfactory point in the parameter space, samples are divided into training and validation subsets during the training process. The performance on the validation subset serves as an indicator to determine whether the iteration should

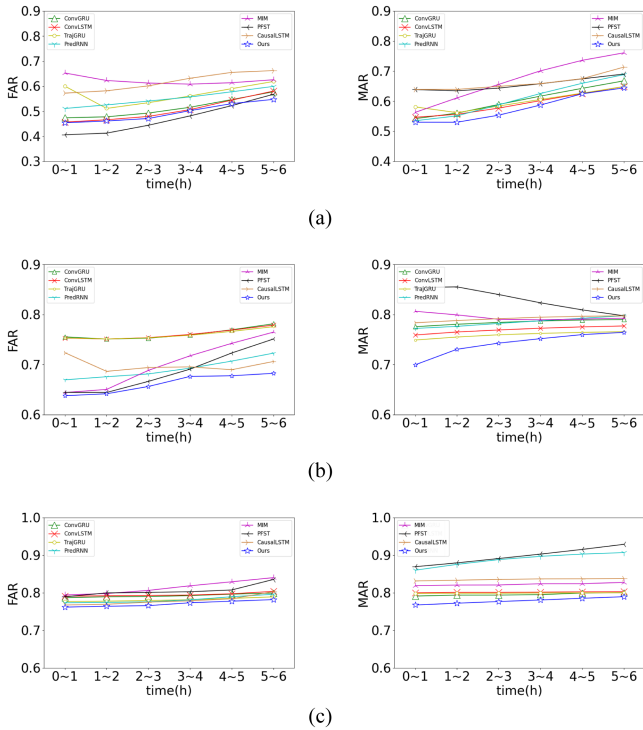


Fig. 9. FAR and MAR for deep learning-based methods on ERA5 regional dataset, WeatherBench dataset, and RainBench dataset.

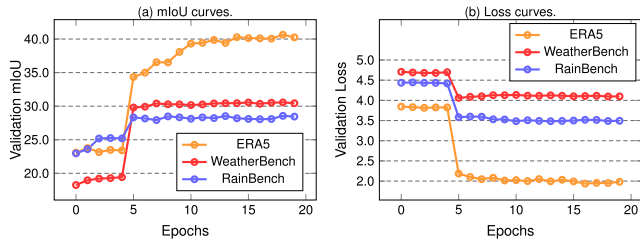


Fig. 10. (a) mIoU curves and (b) loss curves of the STIN on the ERA5, WeatherBench, and RainBench datasets.

continue or be terminated. Fortunately, gradient backpropagation, developed within the framework of gradient descent, is a powerful technique for achieving this objective. To validate this, we plot the mIoU curves of the validation datasets, as shown in Fig. 10(a). Notably, the mIoU curves consistently increase and eventually stabilizes. Furthermore, with the assistance of the Adam optimizer, the training process consistently converges to a local optimum, ensuring stability. Consequently, the value of the loss function consistently decreases until it stabilizes. To verify this, we illustrate the loss evolution during the training process, as depicted in Fig. 10(b). Notably, the loss consistently decreases and eventually stabilizes.

D. Ablation Experiments

1) *Different Backbone*: To verify the capability of STACNN on spatiotemporal variance, the STACNN are replaced by ResNet50 [65] and RedNet50. It should be noticed that ResNet50 and RedNet50 are modified to adapt the fusion strategy with STACNN. Meanwhile, considering the input of STACNN, including modality and solar elevation angle, the

TABLE V
COMPARISON OF DIFFERENT BACKBONES ON THE ERA5 DATASET

Time	Model	TS	IoU	MAR	FAR
3	ResNet50	20.58	40.68	57.07	53.27
	ResNet50+ST	20.42	40.33	61.73	48.38
	ResNet50+SA	20.80	40.21	61.19	49.21
	RedNet50	20.89	40.19	60.32	50.90
	RedNet50+ST	21.11	40.42	59.30	53.87
	RedNet50+SA	20.71	39.78	56.71	53.10
	STACNN w/o ST	21.70	40.95	56.82	53.23
	STACNN+SA	19.61	39.06	63.16	47.72
	STACNN	22.06	41.76	56.35	45.58
6	ResNet50	18.01	34.88	68.68	56.58
	ResNet50+ST	17.26	34.78	68.93	57.32
	ResNet50+SA	17.32	35.05	67.98	59.02
	RedNet50	18.03	35.39	67.90	59.98
	RedNet50+ST	17.88	35.31	68.01	61.07
	RedNet50+SA	17.95	35.88	67.88	57.05
	STACNN w/o ST	18.01	35.63	68.09	60.19
	STACNN+SA	17.68	34.32	67.52	60.89
	STACNN	18.85	36.06	67.42	54.47

The bold values indicate the best performance.

TABLE VI
COMPARISON OF DIFFERENT DYNAMIC LAYERS ON THE ERA5 DATASET

Time	Model	Params(M)	MACs(G)	TS	IoU	MAR	FAR
3	DyConv	109.87	753.64	20.99	40.63	60.23	47.04
	STACConv	64.85	758.17	22.06	41.76	56.35	45.58
6	DyConv	109.87	753.64	17.81	34.56	69.91	56.61
	STACConv	64.85	758.17	18.85	36.06	67.42	54.47

The bold values indicate the best performance.

original ResNet50 and RedNet50 are modified to adapt to the modification of the input tensor, which is the concatenation of modality and solar elevation angle. As given in Table V, a steady improvement could be observed that STACNN yields better results than other backbones. These observations prove that the performance gain does not come from the auxiliary input tensor but benefits from the kernels encoded in spatiotemporal variance. Furthermore, despite the fact that STACNNs and RedNet share the kernels of location variant, STACNN imports spatiotemporal information to generate kernels other than features from itself, leading to an improvement from 39.1% to 40.4% without more parameters and computation complexity. To evaluate the influence of spatiotemporal information on the backbone's performance, we remove the spatiotemporal information and replace it with spatial attention. As illustrated in Table V, the performance of these alternative approaches drops when compared with the incorporation of spatiotemporal information. This substantiates the significance of spatiotemporal-specific kernels generated based on spatiotemporal information for the modeling of meteorological features.

2) *Different Dynamic Layer*: To investigate the effectiveness of the STACConv, we replace it with the dynamic convolution (DyConv) proposed in DY-CNNs [43]. As indicated in Table VI, the performance of STACConv outperforms than Dyconv. We argue that the reason behind this phenomenon stems from DyConv being subjected to spatial-agnostic, which is not completely fit to the modeling meteorological data features.

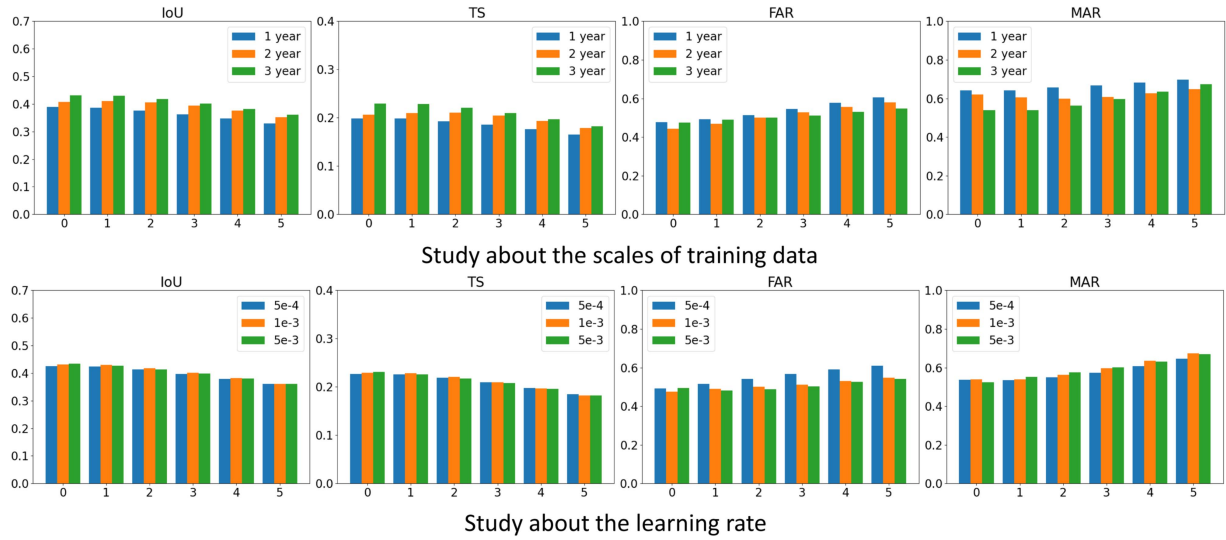


Fig. 11. Hyperparameter experiments about several hyperparameters, including the scales of the training dataset and the learning rate.

TABLE VII

COMPARISON OF DIFFERENT FUSION STRATEGIES ON THE ERA5 DATASET

Time	Model	TS	IoU	MAR	FAR
3	Nonshuffle	19.63	38.57	61.80	53.12
	Channel shuffle	22.06	41.76	56.35	45.58
6	Nonshuffle	19.63	38.57	61.80	53.12
	Channel shuffle	18.85	36.06	67.42	54.47

The bold values indicate the best performance.

TABLE VIII

COMPARISON OF DIFFERENT RNN LAYER ON THE ERA5 DATASET

Time	Model	Params	MACs	TS	IoU	MAR	FAR
3	ConvLSTM	64.85	758.17	22.06	41.76	56.35	45.58
	ConvGRU	62.22	748.25	21.24	40.40	60.07	48.51
	TrajGRU	62.36	733.36	21.72	40.99	56.60	51.65
6	ConvLSTM	64.85	758.17	18.85	36.06	67.42	54.47
	ConvGRU	62.22	748.25	17.68	35.22	69.98	52.66
	TrajGRU	62.36	733.36	17.57	35.17	69.87	54.68

The bold values indicate the best performance.

3) *Different Fusion Strategies*: In the multimodal dimension, we probe the effect of fusion strategies. Specifically, the channel shuffle strategy is compared with the nonshuffle approach. The result of the comparisons is illustrated in Table VII. Benefitting from the channel exchanging process, the channel shuffle strategy achieves better performance than the nonshuffle approach with almost the same complexity and parameters. This observation proves that the interweaves between modality and channel information provide abundant features for modality fusion.

4) *Different RNN Layer*: In Table VIII, we evaluate the impact of different RNN layers on the ERA5 dataset. For fair comparisons, each sequence model's hidden states and convolutional kernels are set to 128 and 3, respectively. It is observed that STIN with ConvLSTM layer outperforms others in TS and IoU. Moreover, STIN with a different RNN layer outperforms corresponding RNN networks. For ConvLSTM, ConvGRU, and

TABLE IX

COMPARISONS OF DIFFERENT SPATIOTEMPORAL INFORMATION

Time	Spatiotemporal Information	TS \uparrow	IoU \uparrow	MAR \downarrow	FAR \downarrow
3	h_s, Φ, δ	20.42	39.62	58.94	50.12
	solar elevation angle	22.06	41.76	56.35	45.58
6	h_s, Φ, δ	18.81	35.62	67.89	56.62
	solar elevation angle	18.85	36.06	67.42	54.47

The bold values indicate the best performance.

TrajGRU, STIN brings 1.9%, 1.4%, and 3.2% improvement on IoU. These improvements further prove that spatial feature extraction is more significant than modeling temporal dependencies in our tasks.

5) *Different Spatiotemporal Information*: To investigate the effectiveness of the solar elevation angle, we replace it with the directly input concatenated h_s, Φ , and δ . The results are presented in Table IX and our approach surpasses the input concatenation evidently. We argue that the solar elevation angle serves as a nonlinear mapping of temporal and spatial information, effectively amalgamating these two domains. In contrast, concatenation necessitates the model to first align temporal and spatial information in the feature space considering heterogeneity, rendering it less efficient than the solar elevation angle.

6) *Effectiveness of Each Component in the STACNN Model*: Ablation experiments were conducted to assess the effectiveness of each component in the STACNN model. To investigate the impact of the proposed STACConv and STACNN, the STACConv was replaced with a normal convolutional layer and an involutorial layer, and the structure of the STACNN was modified to include both a single branch and a multibranch. The results are presented in Table X. STACConv performs better than both the normal convolutional layer and the involutorial layer with the same structure, which can be attributed to the suitability of STACConv for meteorological data with spatiotemporal variance. Similarly, the STACNN outperforms the others with the same convolutional layer by effectively fusing multimodal features.

TABLE X
ABLATION STUDIES ON THE EFFECTIVENESS OF EACH COMPONENT OF
STACNN ON THE ERA5 DATASET

Convs	Structure	TS	IoU	MAR	FAR
Normal Conv	Single branch	19.64	38.57	63.70	50.92
Normal Conv	Multibranch	19.09	37.94	63.71	50.76
Normal Conv	STACNN	20.02	39.05	60.21	51.58
Involution	Single branch	20.27	39.02	59.34	53.57
Involution	Multibranch	20.28	39.07	60.94	53.23
Involution	STACNN	20.32	39.20	59.32	54.22
STACnv	Single branch	20.43	39.18	61.34	52.96
STACnv	Multibranch	20.34	39.26	60.91	51.83
STACnv	STACNN	21.11	40.37	59.16	50.61

The bold values indicate the best performance.

E. Hyperparameter Experiments

In this section, several ablation experiments are conducted to verify the contributions of each hyperparameter. All ablation experiments are conducted on the ERA5 dataset. For each experiment, we report the mean value of the overall forecasting sequence of every modality. Other hyperparameters inherit directly from the statements in Section V-A except for the hyperparameter of the experiment.

Study about the scales of training data: In order to explore the impact of the scales of training data, we train the proposed model with different years and report the result in Fig. 11. As the scales of training data drop, the performance of the proposed model decrease due to the lack of enough samples to improve the generalization.

Study about the learning rates: To analyze the sensitivity of the learning rate, we train the proposed model with a different learning rate, and the result is illustrated in Fig. 11. The performance of the proposed model varies slightly as the learning rate changes except for the FAR, which indicates that the proposed method is robust for learning. When the learning rate equals $1e^{-3}$, the IoU and TS are slightly larger than others. Thus, we choose the initial setting of the learning rate equal to $1e^{-3}$.

VI. CONCLUSION

In this article, STIN is proposed for precipitation nowcasting based on historical meteorological data. Our model is divided into three parts. In the first part, the STACNN is applied to capture the correlation between different modalities and scales. The second part employs the encoder–decoder framework to infer the temporal dependencies. Moreover, our model considers the imbalance in precipitation, and dice loss is employed to provide supervision information. The performance of the STACNN is evaluated for predicting hourly precipitation intensity in the next six hours, utilizing multimodal meteorological data from the preceding six hours. The model’s performance is comprehensively analyzed using four metrics: IoU, TS, FAR, and MAR. Among them, TS and IoU metrics assess the overlap between the predicted and actual precipitation areas. FAR quantifies the proportion of forecasted precipitation areas where no actual precipitation occurs, whereas MAR signifies the proportion of missed precipitation areas within the genuine precipitation region. These experiments on regional and global meteorological datasets demonstrate that our model outperforms all the

baselines. For future work, we will extend our method to address the problem of long-term prediction and focus on capturing temporal dependencies effectively.

REFERENCES

- [1] K. Chris and H. George, “Global precipitation measurement,” *Meteorological Appl.*, vol. 18, no. 3, pp. 334–353, 2011.
- [2] H. Y. Arthur et al., “The global precipitation measurement mission,” *Bull. Amer. Meteorological Soc.*, vol. 95, no. 5, pp. 701–722, 2014.
- [3] B. Peter, T. Alan, and B. Gilbert, “The quiet revolution of numerical weather prediction,” *Nature*, vol. 525, no. 7567, pp. 47–55, 2015.
- [4] P. Lynch, “The origins of computer weather prediction and climate modeling,” *J. Comput. Phys.*, vol. 227, no. 7, pp. 3431–3444, 2008.
- [5] H. Kristine, U. Louis, W. K. Eugenia, C. Kenneth, and M. Lauren, “50th anniversary of operational numerical weather prediction,” *Bull. Amer. Meteorological Soc.*, vol. 88, no. 5, pp. 639–650, 2007.
- [6] S. Juanzhen et al., “Use of NWP for nowcasting convective precipitation: Recent progress and challenges,” *Bull. Amer. Meteorological Soc.*, vol. 95, no. 3, pp. 409–426, 2014.
- [7] V. Peter, K. Peter, F. Andreas, F. S. Andreas, and G. Tilmann, “Skill of global raw and postprocessed ensemble predictions of rainfall over northern tropical Africa,” *Weather Forecasting*, vol. 33, no. 2, pp. 369–388, 2018.
- [8] W. Wangchun and W. Waikin, “Operational application of optical flow techniques to radar-based rainfall nowcasting,” *Atmosphere*, vol. 8, no. 3, 2017, Art. no. 48.
- [9] B. AC, “The size distribution of raindrops,” *Quart. J. Roy. Meteorological Soc.*, vol. 76, no. 327, pp. 16–36, 1950.
- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.
- [11] S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [12] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
- [13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. Conf. North Amer. Chap. Assoc. Comput. Linguistics: Human Lang. Technol.*, 2018, pp. 4171–4186.
- [14] A. Vaswani et al., “Attention is all you need,” in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [15] G. Hinton et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
- [16] A. Graves, A. R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013, pp. 6645–6649.
- [17] S. Agrawal, L. Barrington, C. Bromberg, J. Burge, C. Gazen, and J. Hickey, “Machine learning for precipitation nowcasting from radar images,” 2019, *arXiv:1912.12132*.
- [18] P. Grönquist et al., “Deep learning for post-processing ensemble weather forecasts,” *Philos. Trans. Roy. Soc. A*, vol. 379, no. 2194, 2021, Art. no. 20200092.
- [19] E. Hernández, V. Sanchez-Anguix, V. Julian, J. Palanca, and N. Duque, “Rainfall prediction: A deep learning approach,” in *Proc. Int. Conf. Hybrid Artif. Intell. Syst.*, 2016, pp. 151–162.
- [20] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph WaveNet for deep spatial-temporal graph modeling,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 1907–1913.
- [21] T. Guoqiang, M. Yingzhao, L. Di, Z. Lingzhi, and H. Yang, “Evaluation of GPM day-1 imerg and TMPA version-7 legacy products over mainland China at multiple spatiotemporal scales,” *J. Hydrol.*, vol. 533, pp. 152–167, 2016.
- [22] W. Cunguang, T. Guoqiang, H. Zhongying, G. Xiaolin, and H. Yang, “Global intercomparison and regional evaluation of GPM imerg version-03, version-04 and its latest version-05 precipitation products: Similarity, difference and improvements,” *J. Hydrol.*, vol. 564, pp. 342–356, 2018.
- [23] H. Lin, Z. Gao, Y. Xu, L. Wu, L. Li, and S. Z. Li, “Conditional local convolution for spatio-temporal meteorological forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 7470–7478.

- [24] T. Wilson, P. Tan, and L. Luo, "A low rank weighted graph convolutional approach to weather prediction," in *Proc. IEEE Int. Conf. Data Mining*, 2018, pp. 627–636.
- [25] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Neural Inf. Process. Syst.*, 2015, pp. 802–810.
- [26] W. WC and W. WK, "Application of optical flow techniques to rainfall nowcasting," in *Proc. Conf. Severe Local Storms*, 2014.
- [27] V. Lebedev et al., "Precipitation nowcasting with satellite imagery," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2680–2688.
- [28] T. Baltrusaitis, C. Ahuja, and L. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 423–443, Feb. 2019.
- [29] Y. Cui, Y. Song, C. Sun, A. Howard, and S. J. Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4109–4118.
- [30] R. Stephan, D. Peter, D. S. Sebastian, W. Jonathan, A. M. Soukayna, and T. Nils, "WeatherBench: A benchmark data set for data-driven weather forecasting," *J. Adv. Model. Earth Syst.*, vol. 12, no. 11, 2020, Art. no. e2020MS002203.
- [31] C. S. de Witt et al., "RainBench: Towards data-driven global precipitation forecasting from satellite imagery," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 14902–14910.
- [32] X. Jia, B. D. Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 667–675.
- [33] A. Diba, V. Sharma, L. V. Gool, and R. Stiefelhagen, "DynamoNet: Dynamic action and motion network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6192–6201.
- [34] B. Mildenhall, J. T. Barron, J. Chen, D. Sharlet, R. Ng, and R. Carroll, "Burst denoising with kernel prediction networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2502–2510.
- [35] N. Ma, X. Zhang, J. Huang, and J. Sun, "WeightNet: Revisiting the design space of weight networks," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 776–792.
- [36] Y. Li et al., "Revisiting dynamic convolution via matrix decomposition," in *Proc. Int. Conf. Learn. Representations*, 2021.
- [37] D. Li et al., "Involution: Inverting the inference of convolution for visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12321–12330.
- [38] J. Zhou, V. Jampani, Z. Pi, Q. Liu, and M.-H. Yang, "Decoupled dynamic filter networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6647–6656.
- [39] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
- [40] F. Milletari, N. Navab, and S. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis.*, 2016, pp. 565–571.
- [41] C. Bai, D. Zhao, M. Zhang, and J. Zhang, "Multimodal information fusion for weather systems and clouds identification from satellite images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 7333–7345, Aug. 2022, doi: [10.1109/JSTARS.2022.3202246](https://doi.org/10.1109/JSTARS.2022.3202246).
- [42] D. Zhao, Q. Wang, J. Zhang, and C. Bai, "Mine diversified contents of multispectral cloud images along with geographical information for multilabel classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–15, Apr. 2023, doi: [10.1109/TGRS.2023.3270204](https://doi.org/10.1109/TGRS.2023.3270204).
- [43] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11030–11039.
- [44] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 833–851.
- [45] P. Zhuang, Q. Liu, and X. Ding, "Pan-GGF: A probabilistic method for pan-sharpening with gradient domain guided image filtering," *Signal Process.*, vol. 156, pp. 177–190, 2019.
- [46] P. Guo, P. Zhuang, and Y. Guo, "Bayesian pan-sharpening with multiorder gradient-based deep network constraints," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 950–962, Mar. 2020, doi: [10.1109/JSTARS.2020.2975000](https://doi.org/10.1109/JSTARS.2020.2975000).
- [47] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7077–7087.
- [48] A. Nagrani, S. Yang, A. Arnab, A. Jansen, C. Schmid, and C. Sun, "Attention bottlenecks for multimodal fusion?" in *Proc. Neural Inf. Process. Syst.*, 2021, pp. 14200–14213.
- [49] Y. Gao, M. Zhang, J. Wang, and W. Li, "Cross-scale mixing attention for multisource remote sensing data fusion and classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–15, Mar. 2023.
- [50] Y. Gao et al., "Hyperspectral and multispectral classification for coastal wetland using depthwise feature interaction network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, Jul. 2021.
- [51] Y. Gao, M. Zhang, W. Li, X. Song, X. Jiang, and Y. Ma, "Adversarial complementary learning for multisource remote sensing classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–13, Mar. 2023.
- [52] M. Zhang, X. Zhao, W. Li, Y. Zhang, R. Tao, and Q. Du, "Cross-scene joint classification of multisource data with multilevel domain adaption network," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 06, 2023, doi: [10.1109/TNNLS.2023.3262599](https://doi.org/10.1109/TNNLS.2023.3262599).
- [53] M. Zhang, W. Li, Y. Zhang, R. Tao, and Q. Du, "Hyperspectral and LiDAR data classification based on structural optimization transmission," *IEEE Trans. Cybern.*, vol. 53, no. 5, pp. 3153–3164, May 2023.
- [54] Y. Wang, F. Sun, D. Li, and A. Yao, "Resolution switchable networks for runtime efficient image recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 533–549.
- [55] J. Yu, L. Yang, N. Xu, J. Yang, and T. S. Huang, "Slimmable neural networks," in *Proc. Int. Conf. Learn. Representations*, 2019.
- [56] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6848–6856.
- [57] Y. Zhang et al., "Skilful nowcasting of extreme precipitation with NowcastNet," *Nature*, vol. 619, pp. 526–532, 2023.
- [58] N. Ballas, L. Yao, C. Pal, and A. C. Courville, "Delving deeper into convolutional networks for learning video representations," in *Proc. Int. Conf. Learn. Representations*, 2016.
- [59] X. Shi et al., "Deep learning for precipitation nowcasting: A benchmark and a new model," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 5622–5632.
- [60] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, "PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 879–888.
- [61] Y. Wang, J. Zhang, H. Zhu, M. Long, J. Wang, and P. S. Yu, "Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Comput. Vis. Found.*, 2019, pp. 9154–9162.
- [62] Y. Wang, Z. Gao, M. Long, J. Wang, and P. S. Yu, "PredRNN: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5110–5119.
- [63] C. Luo, X. Li, and Y. Ye, "PFST-LSTM: A spatiotemporal LSTM model with pseudoflow prediction for precipitation nowcasting," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 843–857, Nov. 2020.
- [64] R. E. Huschke, "Glossary of meteorology," 1959.
- [65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.



Qizhao Jin received the B.S. degree in automation from the Harbin Institute of Technology, Harbin, China, in 2016. He is currently working toward the Ph.D. degree in computer application with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

His research focuses on the theory and application of deep learning.



Xinbang Zhang received the B.S. degree in automation from the University of Northeastern University, Shenyang, China, in 2017. He is currently working toward the Ph.D. degree in pattern recognition and intelligent systems with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

His research interests include auto ML and application of deep learning.



Xinyu Xiao received the B.S. degree in detection guidance and control engineering from the School of Astronautics, Beihang University, Beijing, China, in 2015. He is currently working toward the Ph.D. degree in pattern recognition and intelligent system from the Institute of Automation, Chinese Academy of Sciences, Beijing.

His research interests include deep learning, computation, and language.



Ying wang received the B.S. degree in electronic information engineering from the Nanjing University of Information Science and Technology, Nanjing, China, in 2005, the M.S. degree in computer application from the Nanjing University of Aeronautics and Astronautics, Nanjing, in 2008, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2012.

He is currently an Associate Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision, pattern recognition, and remote sensing image processing.



Gaofeng Meng (Senior Member, IEEE) received the B.S. degree in applied mathematics from Northwestern Polytechnical University, Xi'an, China, in 2002, the M.S. degree in applied mathematics from Tianjin University, Tianjin, China, in 2005, and the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, in 2009.

In 2009, he joined the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, as an Assistant Professor. From May to July 2015, he was a Visiting Scholar with the Delft University of Technology, the Netherlands. From 2016 to 2017, he was a Visiting Scholar with Northwestern University, Evanston, IL, USA. He is currently a Professor of National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include document image processing, computer vision, and machine learning.

Dr. Meng was an Associate Editor for *Neurocomputing* and *Image and Vision Computing*.



Shiming Xiang (Member, IEEE) received the B.S. degree in mathematics from Chongqing Normal University, Chongqing, China, in 1993, the M.S. degree in computational mechanics from Chongqing University, Chongqing, in 1996, and the Ph.D. degree in computer application from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2004.

From 1996 to 2001, he was a Lecturer with the Huazhong University of Science and Technology, Wuhan, China. Until 2006, he was a Postdoctorate Candidate with the Department of Automation, Tsinghua University, Beijing. He is currently a Professor with the National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing. His research interests include pattern recognition, machine learning, and computer vision.



Chunhong Pan (Member, IEEE) received the B.S. degree in automatic control in optical information processing from Tsinghua University, Beijing, China, in 1987, the M.S. degree in optical information processing from the Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, Shanghai, China, in 1990, and the Ph.D. degree in pattern recognition and intelligent system from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2000.

He is currently a Professor with the National Laboratory of Pattern Recognition of Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision, image processing, computer graphics, and remote sensing.