






# SA-JSTN: Self-Attention Joint Spatiotemporal Network for Temperature Forecasting

Lukui Shi , Nanying Liang , Xia Xu , *Graduate Student Member, IEEE*, Tao Li , *Member, IEEE*, and Zhou Zhang 

**Abstract**—The rapid development of remote sensing technology has brought abundant data support for deep learning based temperature forecasting research. However, recently proposed methods usually focus on the temporal relationship among temperature observation information, whereas ignore the spatial positions of different regions. Motivated by the observation that adjacent regions usually present similar temperature trends, in this article, we consider the temperature forecasting as a spatiotemporal sequence prediction problem, and propose a new deep learning model for temperature forecasting, self-attention joint spatiotemporal network (SA-JSTN), which simultaneously captures the spatiotemporal interdependency information. The kernel component of the SA-JSTN is a newly developed spatiotemporal memory (STM) unit, which describes the temporal and spatial models via a unified memory cell. STM is constructed based on the units of the convolutional long short-term memory (ConvLSTM). Instead of using simple convolutions for spatial information extraction, in STM, we improve ConvLSTM by a self-attention module, which has significantly enhanced the global spatial information representation ability of our proposed network. Compared with other deep learning based temperature forecasting methods, SA-JSTN is able to integrate the global spatial correlation into the temperature series prediction problem, and thus present better performance especially in short-term prediction. We have conducted comparison experiments on two typical temperature datasets to validate the effectiveness of our proposed method.

**Index Terms**—Long short-term memory (LSTM), self-attention, spatiotemporal prediction, temperature forecasting.

Manuscript received June 19, 2021; revised August 13, 2021 and September 5, 2021; accepted September 9, 2021. Date of publication September 14, 2021; date of current version September 30, 2021. This work was supported in part by the China Postdoctoral Science Foundation under Grant 2020M670631, in part by the Science and Technology Foundation of State Key Laboratory under Grant 61420020401, in part by the Natural Science Foundation of Hebei province of China under Grant F2020202008, in part by the Key R&D Plan of Tianjin Science and Technology Bureau under Grant 20YFZCGX00490, in part by the National Natural Science Foundation of China under Grant 62001251, and in part by the Scientific and Technological Research Project of Hebei Province Universities and Colleges under Grant ZD2021311. (*Corresponding authors: Xia Xu; Tao Li.*)

Lukui Shi and Nanying Liang are with the School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China, and also with the Hebei Province Key Laboratory of Big Data Calculation, Tianjin 300401, China (e-mail: shilukui@scse.hebut.edu.cn; liangnanying@hotmail.com).

Xia Xu and Tao Li are with the College of Computer Science, Nankai University, Tianjin 300071, China, and also with the Key Laboratory of Pure Mathematics and Combinatorics, Ministry of Education, Tianjin 300071, China (e-mail: xuxia@nankai.edu.cn; litao@nankai.edu.cn).

Zhou Zhang is with the Department of Biological Systems Engineering, University of Wisconsin-Madison, Madison, WI 53706 USA (e-mail: zzhang347@wisc.edu).

Digital Object Identifier 10.1109/JSTARS.2021.3112131

## I. INTRODUCTION

**I**N RECENT years, the rapid development of remote sensing techniques [1]–[3] have brought exciting data and resource support for temperature forecasting problem. Temperature forecasting is to predict the temperature in a local region over a relatively short period of time precisely. Temperature forecasting is of paramount importance in many practical applications [4], [5], such as traffic control, weather forecasting, infectious disease prevention, and environmental monitoring. However, building an accurate temperature forecasting model still remains challenging, because the random behavior of temperature changes is not only dominated by physics regulations but also affected by spatial and temporal changes in local areas.

Existing temperature forecasting algorithms can be roughly divided into two categories: physical-based numerical models and data-driven models. Goldberg *et al.* proposed a series of physics-based numerical methods [6]–[11] to simulate the equations of hydrodynamics and thermodynamics in the process of atmospheric evolution, which predicted the atmospheric motion state and weather phenomena in a period of time in the future. However, numerical models usually spend too much computational resources and time acquiring data and predicting temperature, which leads to low efficiency of prediction. In recent years, with more and more temperature remote sensing data, grid data, and meteorological station observation data [12]–[14] being collected, stored, processed, and disseminated, many researchers have proposed data-driven models. The data-driven approaches learn patterns and relationships from historical observations to further extrapolate future temperatures. The autoregressive moving average models and their variants have proved effectiveness in the application of temperature forecasting [15]–[24], but they can only model linear relationships. However, the temperature data are nonlinear and have a very irregular trend. To solve the problems of data dependence and complex mechanisms in temperature forecasting, algorithms based on deep learning are proposed.

Deep learning algorithms can better model the nonlinear relationship of temperature data. Inspired by biological nervous systems, artificial neural network (ANN) is a powerful tool for modeling nonlinear relationships between independent and dependent variables. Therefore, different types of ANN [25]–[29] are applied in the domain of temperature forecasting, such as multilayer perceptron, recurrent neural network (RNN), long short-term memory (LSTM), and convolutional neural

network (CNN). Zaytar *et al.* presented temperature forecasting methods based on LSTM [30]–[41] to better model the intrinsic relationship of temperature series. However, LSTM usually only considers the temporal correlation between temperatures, while ignoring the spatial correlation of different regions. Temperature forecasting is actually a spatiotemporal sequence prediction problem. Therefore, algorithms based on spatiotemporal correlation are proposed. The concept of spatiotemporal sequence prediction algorithms originated from convolutional long short-term memory (ConvLSTM) [42], which was successfully applied to precipitation nowcasting. Karevan *et al.* [43] defined a spatiotemporal stacked LSTM model for weather forecasting. Nascimento *et al.* [44] proposed a spatiotemporal convolutional sequence to sequence network (STConvS2S), which only applied convolutional layers to learn spatiotemporal correlation of temperature data. However, LSTM tends to model the temporal structure but lacks the ability of capturing visual appearance, while CNN pays more attention to spatial appearance and has a poor ability to capture long-term motion. To alleviate the defects of LSTM and CNN in the prediction task, researchers consider using CNN and RNN [45] together. Wang *et al.* [46] constructed a predictive recurrent neural network (PredRNN), which puts spatial appearances and temporal variations in a unified memory pool for the first time. Yang *et al.* [47] proposed a combined fully connected LSTM and convolution neural network model to improve the prediction accuracy of sea surface temperature.

Although the existing deep learning algorithms can achieve good prediction results, there are still some problems. In fact, the temperature observations in different regions have certain regularity. However, due to the regional fixity, the local temperature changes are not affected in the same way by topography, latitude, and other factors. The convolution operation is usually utilized to capture the spatial correlation of temperature information in deep learning temperature forecasting algorithms, which is local and inefficient. The use of convolution is equivalent to the default that the temperature in different regions is affected in the same way by geographical factors, which will lead to inaccurate prediction results. Therefore, it is very important to consider the global spatial context in the process of temperature forecasting. Long-range spatial dependencies are significant for spatial applications.

In this article, we propose a self-attention joint spatiotemporal network (SA-JSTN) for temperature forecasting, which has certain advantages in predicting sudden temperature changes in local areas by paying attention to global information. The method is inspired by PredRNN, which performs well in the practical application of using radar echo data for precipitation nowcasting. However, PredRNN exploits convolution operation to capture spatial dimension correlation in local neighborhoods, ignoring the influence of geographical factors in different locations on the observed values. Therefore, we propose an SA-JSTN model, which can model spatiotemporal dimensions simultaneously and capture global spatial context information.

The main contributions of this work are as follows.

- 1) We propose a new deep learning model, SA-JSTN, for temperature forecasting, which may have advantage in extracting the spatial dependence between observations

at different regions. SA-JSTN transmits information in both horizontal and vertical directions with a stacked RNN architecture, and simultaneously captures the spatiotemporal correlation of temperature data.

- 2) We develop a new spatiotemporal memory (STM) structure, which integrates the spatiotemporal information into a unified unit. Moreover, different from simple convolutions, STM is able to capture the global spatial contextual information via a self-attention module, which may better describe the interactions between different regions.

The rest of this article is organized as follows. Section II discusses works related to both temperature forecasting and spatiotemporal architectures. Section III introduces the SA-JSTN architecture and STM module. Section IV provides dataset information, network settings, experimental results, corresponding analysis, and discussion. Finally, Section V summarizes this article.

## II. RELATED WORK

Several physics-based numerical models and data-driven models have been applied to historical temperature data to predict the weather conditions. Autoregressive integrated moving average (ARIMA) is a traditional method for time series prediction [48]. Some scholars have studied new methods to improve the prediction accuracy of spatiotemporal series based on deep learning in recent years. The latest development of RNN models [30]–[41] provided useful insights for predicting future temperature series based on historical observations. However, RNN models cannot capture the spatial dependence of the observation results. Spatiotemporal deep learning models can simultaneously capture the context of spatiotemporal dimensions.

ConvLSTM [42] based models are a crucial branch in spatiotemporal sequence prediction. ConvLSTM captured spatial context information through convolution operations on the input-to-state and state-to-state transitions in the LSTM network. ConvLSTM established an end-to-end model for precipitation nowcasting. Based on ConvLSTM, PredRNN [46] and improved predictive recurrent neural network [49] improved predictive performance by introducing additional global memory cells and their reorganization. Memory in memory (MIM) [50] added nonstationary modeling to the ConvLSTM unit and further exploited the differential signal between adjacent recurrent states to update memory cells into nonstationary information and stationary information. STConvS2S [44] only applied the 3-D convolutional layers to learn the spatiotemporal correlation of temperature data. STConvS2S learned the spatiotemporal representation of the input sequence through the temporal block and the spatial block, and controlled the sequence prediction length through the temporal generator block. In contrast, STConvS2S tends to model spatial representation through convolution operations and has a poor ability to capture long-term motion. Although the spatiotemporal prediction algorithms based on ConvLSTM can simultaneously capture spatiotemporal dimension information, the capture of spatial dimension information can only model the local context information through the convolution operation.

The self-attention module [51] was first proposed and applied in natural language processing and has been successfully extended to computer vision [52], [53]. The self-attention module has achieved impressive results in capturing the global spatial context. In this article, we explore a new spatiotemporal prediction learning framework, introduce a self-attention module, and propose a new spatiotemporal memory unit for simultaneous memory of spatiotemporal information in the temperature forecasting process.

### III. METHODOLOGY

In this section, we first state the definition of the spatiotemporal sequence temperature forecasting problem, and then introduce the proposed SA-JSTN architecture and STM module in detail.

#### A. Spatiotemporal Sequence Prediction

The purpose of temperature forecasting based on spatiotemporal sequence is to utilize the previously observed spatiotemporal sequence temperature images to forecast the temperature images of the future period time in a local region. From the perspective of deep learning, temperature forecasting can be regarded as a spatiotemporal sequence forecasting problem, which can be modeled as a sequence-to-sequence problem.

To predict the temperature of a local region, suppose the spatial region is represented by an  $H \times W$  grid, which consists of  $H$  rows and  $W$  columns. Each element in the grid has a temperature value that changes with time. Thus, the temperature observations in a local area at any time can be represented by a tensor  $\mathcal{X} \in \mathbf{R}^{C \times H \times W}$ , where  $\mathbf{R}$  denotes the domain of temperature observations and  $C$  is the number of channels (where  $C$  is 1, similar to a grayscale image). Then, the temperature observations of  $T$  consecutive moments can be expressed as a tensor sequence  $[\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_T]$ . The spatiotemporal sequence forecasting problem is to adopt the previous  $T$  sequences including the current observation to predict the most likely  $T'$  sequences in the future under a given time condition. The process is defined as follows:

$$\begin{aligned} & \left[ \tilde{\mathcal{X}}_{t+1}, \dots, \tilde{\mathcal{X}}_{t+T'} \right] = \\ & \arg \max_{\mathcal{X}_{t+1}, \dots, \mathcal{X}_{t+T'}} p(\mathcal{X}_{t+1}, \dots, \mathcal{X}_{t+T'} | \mathcal{X}_{t-T+1}, \dots, \mathcal{X}_t) \quad (1) \end{aligned}$$

where  $[\tilde{\mathcal{X}}_{t+1}, \dots, \tilde{\mathcal{X}}_{t+T'}]$  is a sequence of forecast result.

#### B. SA-JSTN Architecture

In this section, we give detailed descriptions of the SA-JSTN. Initially, this architecture is inspired by PredRNN. For a spatiotemporal sequence prediction learning system, the spatial appearances and temporal variations should be memorized in a unified memory pool. The design of spatiotemporal long short-term memory (ST-LSTM) realizes the unified modeling of spatiotemporal information. However, the ST-LSTM takes ConvLSTM as the basic building block, which is inevitably limited by the receptive field for spatial appearance modeling. Therefore, we design an SA-JSTN architecture.

SA-JSTN is a deep learning sequence-to-sequence architecture designed for short-term temperature forecasting. As shown in Fig. 1, we apply a stacked RNN architecture that models spatiotemporal sequence prediction tasks. The SA-JSTN structure ensures information to be transmitted vertically over states and horizontally across layers at the same time.

The kernel building block of SA-JSTN is the STM module. The same STM unit is adopted in the temporal dimension, and the parameters are shared in the vertical direction. Stacking multiple different STM units in the spatial dimension, and the parameters are not shared in the horizontal direction. The spatial dimension stacks multilayer STM can extract highly abstract features layer-by-layer and, then, make predictions by mapping them back to the temperature values. The highest spatial state of the previous time step is used to initialize the starting spatial dimension state of the next time step. Formally, the spatial dimension state transition between the two time steps is shown in the following formula:

$$\begin{aligned} \mathcal{M}_t^0 &= \mathcal{M}_{t-1}^n \\ \mathcal{H}_t^0 &= \mathcal{H}_{t-1}^n \end{aligned} \quad (2)$$

where  $\mathcal{M}$  is the spatiotemporal memory,  $\mathcal{H}$  is the hidden layer state, and  $n$  is the number of stacked RNN architecture layers. Note that the initial value  $\mathcal{M}_0^0$  of spatiotemporal memory is initialized with all zeros. The calculation equation of STM architecture with  $n$ -layer stack used in this article is as follows (for  $2 \leq l \leq n$ ):

$$\begin{aligned} [\mathcal{H}_t^1, \mathcal{C}_t^1, \mathcal{M}_t^1] &= \text{STM}_1(\mathcal{X}_t, \mathcal{H}_{t-1}^1, \mathcal{C}_{t-1}^1, \mathcal{M}_t^0) \\ [\mathcal{H}_t^l, \mathcal{C}_t^l, \mathcal{M}_t^l] &= \text{STM}_l(\mathcal{H}_{t-1}^{l-1}, \mathcal{H}_{t-1}^l, \mathcal{C}_{t-1}^l, \mathcal{M}_t^{l-1}) \end{aligned} \quad (3)$$

where input  $\mathcal{X}_t$ , cell outputs  $\mathcal{C}_t^l$ ,  $\mathcal{M}_t^l$ , and hidden state  $\mathcal{H}_t^l$  are tensors in  $\mathbf{R}^{C \times H \times W}$ . Note that the first layer STM is marked as  $\text{STM}_1$ . To solve the problem that ConvLSTM, the basic building block in ST-LSTM structure, has limited ability to capture spatial information, we introduce a self-attention module to model the spatial dimension. The self-attention module can capture global spatial context information and produce more accurate prediction results. The STM module is described in detail below.

#### C. STM Module

The prediction of future time steps can benefit from the relevant characteristics of past time steps; therefore, we construct an STM unit with memory capability. The STM module can model information with the global spatiotemporal receptive field. The structure of STM is shown in Fig. 2. When processing spatiotemporal data, the STM module still utilizes ConvLSTM to capture the relevant information in the temporal dimension. To capture the long-term dependency of context information in the spatial dimension, a self-attention module is added on the basis of ConvLSTM. Finally, the information on temporal and spatial dimensions is jointly utilized to predict temperature.

1) *Temporal Dimension Modeling*: First, according to the state information  $\mathcal{H}_t^{l-1}$  (if  $l = 1$ , then  $\mathcal{H}_t^{l-1} = \mathcal{X}_t$ ) of the upper layer and the previous time state information  $\mathcal{H}_{t-1}^l$ , the convolutional gating structure in ConvLSTM is employed to obtain the

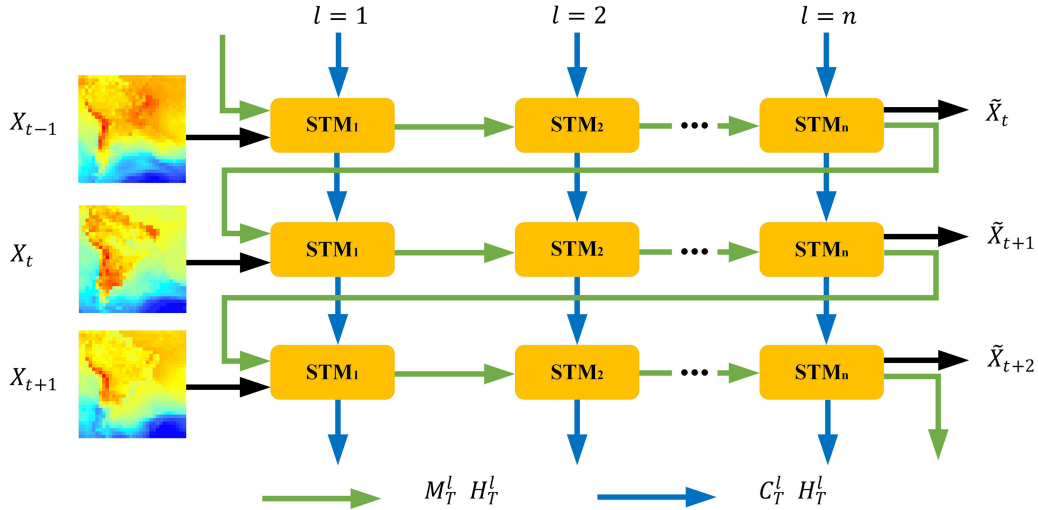


Fig. 1. Graphical illustration of SA-JSTN structure. For simplicity, the input in the figure only shows three consecutive temperature images  $\{\mathcal{X}_{t-1}, \mathcal{X}_t, \mathcal{X}_{t+1}\}$ , which are memorized and transmitted through the STM module to obtain the prediction results. The temporal information flow is transmitted from top to bottom along the blue arrows. The spatial information flow zigzags in the horizontal direction along the green arrows. The number of STM stacking layers is  $n$ .

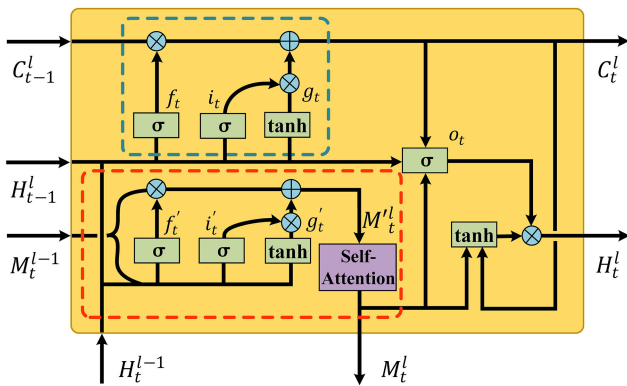


Fig. 2. Graphical illustration of an STM unit structure. The blue dashed box and the red dotted line box, respectively, represent the information transmission process of temporal and spatial dimensions. First, the input is memorized and transferred through the spatiotemporal dimensional modeling units, and then, the joint module is exploited to incorporate the spatiotemporal modeling results to obtain the output state information.

current temperature change information in the temporal domain. Then, a new temporal unit  $C_t^l$  is generated by updating the previous temporal cell state  $C_{t-1}^l$ , as shown in the blue dotted line box in Fig. 2. The temporal dimension information transmission process is shown in the following formula:

$$\begin{aligned}
 g_t &= \tanh(W_{xg} * \mathcal{H}_t^{l-1} + W_{hg} * \mathcal{H}_{t-1}^l + b_g) \\
 i_t &= \sigma(W_{xi} * \mathcal{H}_t^{l-1} + W_{hi} * \mathcal{H}_{t-1}^l + b_i) \\
 f_t &= \sigma(W_{xf} * \mathcal{H}_t^{l-1} + W_{hf} * \mathcal{H}_{t-1}^l + b_f) \\
 C_t^l &= f_t \circ C_{t-1}^l + i_t \circ g_t
 \end{aligned} \quad (4)$$

where  $\sigma$  is the sigmoid activation function, ‘\*’ and ‘ $\circ$ ’ denote the convolution operator and the Hadamard product, respectively, and  $g_t$ ,  $i_t$ , and  $f_t$  are tensors in  $\mathbf{R}^{C \times H \times W}$ .

2) *Spatial Dimension Modeling*: First, according to  $\mathcal{H}_t^{l-1}$  and the cell state  $\mathcal{M}_t^{l-1}$  of the spatial domain, ConvLSTM is exploited to obtain the temperature change information of the current layer. Then, it is utilized to correct the spatial cell state  $\mathcal{M}_t^{l-1}$  of the previous layer to generate a new spatial unit  $\mathcal{M}_t^l$ . Finally, the self-attention module is applied to pay attention to the spatial information  $\mathcal{M}_t^l$  obtained by the ConvLSTM operation, so that the network can focus on the correlation among all pixels in the same context, as shown in the content of the red dotted line box in Fig. 2. The information transmission process of the spatial dimension is shown in the following formula:

$$\begin{aligned}
 g'_t &= \tanh(W'_{xg} * \mathcal{H}_t^{l-1} + W'_{mg} * \mathcal{M}_t^{l-1} + b'_g) \\
 i'_t &= \sigma(W'_{xi} * \mathcal{H}_t^{l-1} + W'_{mi} * \mathcal{M}_t^{l-1} + b'_i) \\
 f'_t &= \sigma(W'_{xf} * \mathcal{H}_t^{l-1} + W'_{mf} * \mathcal{M}_t^{l-1} + b'_f) \\
 \mathcal{M}_t^l &= f'_t \circ \mathcal{M}_t^{l-1} + i'_t \circ g'_t \\
 \mathcal{M}_t^l &= \text{SA}(\mathcal{M}_t^l)
 \end{aligned} \quad (5)$$

where  $g'_t$ ,  $i'_t$ , and  $f'_t$  are tensors in  $\mathbf{R}^{C \times H \times W}$ , and  $\text{SA}(\cdot)$  represents self-attention operation that is illustrated in Fig. 3.

Since the self-attention [51] module was proposed, it has attracted more and more attention because of its flexibility in parallel computing and long-term dependency modeling. The self-attention module has also been widely adopted to a single context due to its good effectiveness in recent years. Connections between elements in all positions are established by calculating the weight of attention. In other words, the query, key, and value defined later in this section, all come from the same context. Formally, given an input  $\mathcal{M}_t^l$ , the state of the output layer is constructed by paying attention to the state of the input layer. Specifically, the input layer  $\mathcal{M}_t^l$  is first converted to query  $\mathbf{Q}$ ,



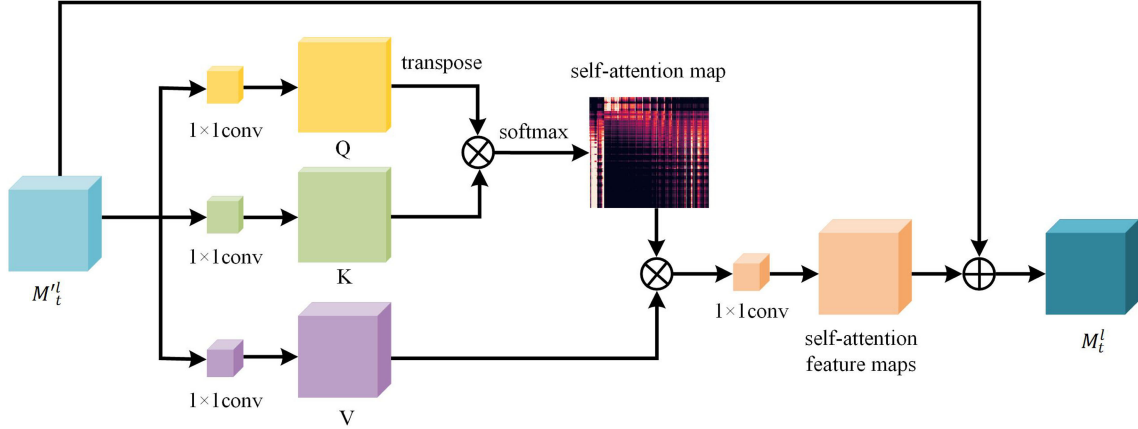


Fig. 3. Working process of the self-attention module. First, input  $\mathcal{M}_t^l$  is converted to query  $\mathbf{Q}$ , key  $\mathbf{K}$ , and value  $\mathbf{V}$  by  $1 \times 1$  convolution operation. Then, the attention matrix is obtained from  $\mathbf{Q}$  and  $\mathbf{K}$  to act on  $\mathbf{V}$ , and self-attention feature maps of the same dimension as the input are generated through  $1 \times 1$  convolution. Finally, the input and self-attention feature maps are added to obtain the cell state  $\mathcal{M}_t^l$  containing global context information.

key  $\mathbf{K}$ , and value  $\mathbf{V}$ , which is described as

$$\begin{bmatrix} \mathbf{Q} \\ \mathbf{K} \\ \mathbf{V} \end{bmatrix} = \begin{bmatrix} W_Q \\ W_K \\ W_V \end{bmatrix} \mathcal{M}_t^l \quad (6)$$

where  $\{W_Q, W_K, W_V\}$  are trainable  $1 \times 1$  convolution parameter matrices. The output  $\mathcal{M}_t^l$  is expressed as follows:

$$\mathcal{M}_t^l = \mathcal{M}_t^l + W_O * \mathbf{O} \quad (7)$$

where

$$\mathbf{O} = \text{ATT}(\mathbf{Q}, \mathbf{K}) \mathbf{V} \quad (8)$$

where  $\text{ATT}(\cdot)$  is an attention model, which can be expressed as the following formula:

$$\text{ATT}(\mathbf{Q}, \mathbf{K}) = \text{softmax} \left( \frac{\mathbf{Q}^T \mathbf{K}}{\sqrt{d}} \right) \quad (9)$$

where  $\sqrt{d}$  is the scale factor.

3) *Joint Mechanism*: In the joint mechanism of the STM unit, the shared output gate is employed to seamlessly combine the memory information of the temporal and spatial memories. The final hidden state of the unit depends on the spatiotemporal memory after the joint. The joint mechanism connects the memory information from the horizontal and vertical directions together. Then, it applies a  $1 \times 1$  convolutional layer to reduce the dimension, so that the hidden state  $\mathcal{H}_t^l$  has the same dimension as  $\mathcal{C}_t^l$  and  $\mathcal{M}_t^l$ . By incorporating spatial with temporal dimension information in a unified unit, it makes the spatiotemporal sequence temperature image prediction more effective. The joint module finally generates the intermediate prediction of the next STM unit input or constructs the final prediction frame. The formula is as follows:

$$\begin{aligned} o_t &= \sigma(W_{x_o} * \mathcal{H}_t^{l-1} + W_{h_o} * \mathcal{H}_{t-1}^l + W_{c_o} * \mathcal{C}_t^l \\ &\quad + W_{m_o} * \mathcal{M}_t^l + b_o) \\ \mathcal{H}_t^l &= o_t \circ \tanh(W_{1 \times 1} * [\mathcal{C}_t^l, \mathcal{M}_t^l]). \end{aligned} \quad (10)$$

In summary, the SA-JSTN model improves the ST-LSTM basic memory unit in PredRNN and proposes a new spatiotemporal joint module STM. The pseudocode of SA-JSTN is shown in Algorithm 1. The spatiotemporal information is simultaneously extracted and memorized in a unified memory pool by (4), (5), and (10). The STM module enhances the ability to capture global context information and breaks the limitation that ConvLSTM can only capture local context information in spatial modeling.

#### IV. EXPERIMENT

We conduct experiments on two temperature datasets to verify the proposed architecture. One is the CFSR public meteorological dataset, and the other is the data provided by the Hebei Meteorological Bureau of China (HMBC dataset). In the experiment, the L1 loss is applied as the data loss term, and the L2 norm is applied as the penalty term. By comprehensively considering the training, verification, and test losses, we set the hyperparameter  $\lambda$  before the penalty term to 1.0 to balance the data loss term and the penalty term. The SA-JSTN model is optimized with L1+L2 loss. An ADAM optimizer is used for training, and the initial learning rate is set to  $10^{-4}$ . The training process is stopped after 80000 iterations. The batch size of each iteration is set to eight. All experiments are performed on the Nvidia GeForce GTX1080 GPU server with 8 GB of RAM and implemented in PyTorch. We first introduce datasets and evaluation metrics. Then, we describe the experimental results and corresponding analysis.

##### A. Datasets and Evaluation Metrics

1) *CFSR Dataset*: It is one of the latest global reanalysis climate datasets and has been widely exploited in climate change research. Atmospheric reanalysis data are driven by a variety of data, and it requires strict quality control. In practice, historical observation data are obtained by applying data assimilation techniques and numerical prediction models. Recently, a new generation of historical reanalysis data is suitable for climate pattern research due to its high spatial resolution. The CFSR dataset

---

**Algorithm 1** SA-JSTN Algorithm
 

---

**Input:**

Data: Training dataset TRAIN, validation dataset VAL, test dataset TEST;

Parameter: Input sequence length  $T$ , total sequence length  $T'$ , the number of stacked layers  $n$ , the number of hidden states  $H$ , the size of the convolution kernel  $K$ , the initial learning rate lr, the maximum number of training iterations  $I$ ;

**Output:**

Prediction result of the spatiotemporal sequence temperature images in the future;

- 1: Initialize all trainable weights  $W$  for network;
  - 2: **for** itr = 1 to  $I$  **do**
  - 3:   **for**  $t = 1$  to  $T'-1$  **do**
  - 4:     **for**  $l = 1$  to  $n$  **do**
  - 5:       Generate a new temporal unit  $C_t^l$ , based on (4);
  - 6:       Generate a new spatial unit  $M_t^l$ , based on (5);
  - 7:       The joint module generates the intermediate prediction of the next STM unit input or constructs the final prediction frame, based on (10);
  - 8:     **end for**
  - 9:     Generate forecast sequence;
  - 10:    Update weights  $W$  of network through L1+L2 loss;
  - 11:   **end for**
  - 12:   **if** itr % 5000 = 0 **then**
  - 13:     save the training model and use the VAL for verification;
  - 14:   **end if**
  - 15: **end for**
  - 16: Choose the model that performs best in the VAL to generate the forecast sequence in the TEST.
- 

contains high-resolution global land and ocean data, including spatial coordinates (latitude and longitude), a spatial resolution of  $0.5^\circ$  (i.e., the area of each grid cell is  $0.5^\circ \times 0.5^\circ$ ), some meteorological variables (such as air temperature and humidity), and sampling frequency at 6-h intervals. In the experiment, to facilitate comparison with other methods, we adopt a subset of the CFSR dataset. The air temperature observation is from January 1979 to December 2015, covering the longitude and latitude ranges of  $80^\circ\text{W}$ – $25^\circ\text{W}$  and  $8^\circ\text{N}$ – $54^\circ\text{S}$ , respectively, as shown in Fig. 4. To adapt to the GPU memory, we scale down the grid size to  $32 \times 32$  ( $H \times W$ ) pixels. To verify the prediction ability of the SA-JSTN model for long sequences, two prediction strategies were adopted. The first strategy uses a sliding window with a width of ten frames to slice the consecutive images. Therefore, each sequence consists of ten consecutive frames, five of which are input frames ( $T$ ), and others are prediction frames ( $T'$ ). The second strategy uses a 20-frame sliding window to slice consecutive images. Therefore, each sequence consists of 20 consecutive frames, five input frames ( $T$ ), and 15 prediction frames ( $T'$ ). The number of sequence samples for the two

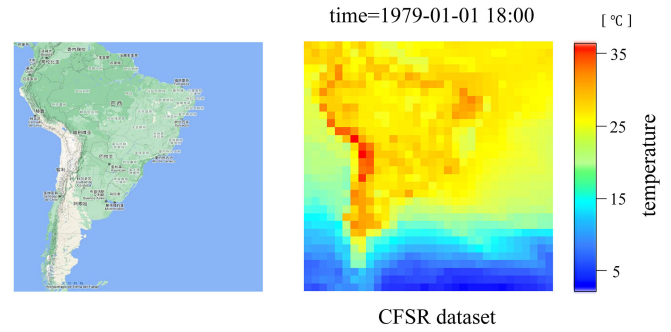


Fig. 4. Left: The topographic map of the CFSR dataset spatial coverage area. Right: The temperature image of CFSR dataset, and the value of each pixel represents the temperature.

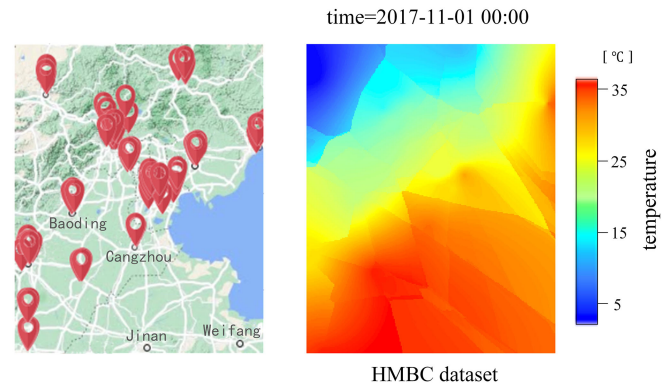


Fig. 5. Left: The topographic map of the HMBC dataset spatial coverage area, where the red dots are marked as site locations. Right: The temperature image of the HMBC dataset obtained by the Kriging interpolation method, and the value of each pixel represents the temperature.

strategies are 54047 and 54037, respectively. In the experiment, the division of sequence samples is based on time. We divide the sequence samples into nonoverlapping training, validation, and test set according to the proportion of 60%, 20%, and 20%.

2) *HMBC Dataset*: It is the meteorological data provided by the HMBC, covering 1591 meteorological stations in China. The data include the index number of the station, 36 meteorological elements (such as temperature and wind speed), and sampling frequency at 3-h intervals. In the experiment, we select the temperature data of 87 meteorological stations in Hebei Province and its surrounding areas to make a spatiotemporal sequence temperature image prediction dataset. The temperature observation time is from November 2017 to March 2018 and from November 2018 to March 2019, covering the longitude and latitude ranges of  $114^\circ\text{E}$ – $120^\circ\text{E}$  and  $36^\circ\text{N}$ – $41^\circ\text{N}$ , respectively. For the preprocessing, we first query the spatial coordinates (latitude and longitude) of the meteorological stations. According to the spatial coordinates, we use the Kriging interpolation method to interpolate, and process the temperature data of 87 meteorological stations into grid temperature data with a size of  $311 \times 251$  ( $H \times W$ ) pixels, as shown in Fig. 5. Since then, the temperature data are randomly clipped into a patch with the size of  $100 \times 100$  ( $H \times W$ ) pixels to fit the GPU memory. For the HMBC dataset, we use a 16-frame sliding window to slice

TABLE I  
COMPARATIVE RESULTS WITH DIFFERENT METHODS ON TEMPERATURE DATASETS

Dataset	Horizon	Metric	ARIMA	ConvLSTM	PredRNN	MIM	STConvS2S-C	STConvS2S-R	SA-JSTN (ours)
CFSR	$T = 5, T' = 5$	MAE	1.9073	1.1805	1.1736	1.1623	0.9180	0.8564	<b>0.8422</b>
		RMSE	2.1935	1.7304	1.6859	1.6207	1.3326	1.2630	<b>1.2273</b>
		FLOPs	—	<b>14.78G</b>	30.13G	46.69G	35.35G	28.91G	30.24G
		Test time/samples	—	<b>0.013s</b>	0.026s	0.053s	0.042s	0.037s	0.033s
HMBC	$T = 5, T' = 15$	MAE	1.9279	1.5071	1.4238	1.4202	1.2743	1.2365	<b>1.1871</b>
		RMSE	2.2755	2.1816	2.0294	2.0233	1.8658	1.8137	<b>1.7446</b>
		FLOPs	—	<b>44.34G</b>	90.39G	144.09G	97.99G	91.55G	90.71G
		Test time/samples	—	<b>0.021s</b>	0.058s	0.082s	0.061s	0.064s	0.060s
HMBC	$T = 8, T' = 8$	MAE	3.7845	2.8109	2.7688	2.7357	2.6109	2.4997	<b>2.1581</b>
		RMSE	4.0720	3.3111	3.2749	3.2631	3.2403	3.0664	<b>2.5857</b>
		FLOPs	—	<b>230.96G</b>	470.79G	729.64G	528.74G	465.78G	472.45G
		Test time/samples	—	<b>0.135s</b>	0.263s	0.357s	0.291s	0.305s	0.289s
		Params	—	<b>2.89M</b>	5.88M	9.12M	6.12M	6.51M	5.91M

The boldface values represent the best results of the five evaluation indicators MAE, RMSE, FLOPs, Params, and Test time/samples on different data sets.

the images. Therefore, each sequence consists of 16 consecutive frames, with eight input frames ( $T$ ) and eight prediction frames ( $T'$ ). The number of sequence samples is 2363, and the same division strategy as the CFSR dataset is adopted.

3) *Evaluation Metrics*: To evaluate the SA-JSTN architecture, we applied five evaluation indicators, namely MAE, RMSE, FLOPs, Params, and Test time/samples.

The formulas of MAE and RMSE are, respectively, shown as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\tilde{x}_i - x_i| \quad (11)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \tilde{x}_i)^2} \quad (12)$$

where  $n$  is the number of test samples, and  $x_i$  and  $\tilde{x}_i$  are the real and predicted values, respectively. In (11), MAE can better reflect the actual situation of the predicted value error. In (12), RMSE is very sensitive to large or small errors and can well reflect the precision of measurement. The lower the value of MAE and RMSE, the better the prediction effect.

FLOPs, Params, and Test time/samples are indicators to evaluate the computational efficiency of the model. FLOPs is floating point of operations, which represents the amount of calculation and is applied to measure the complexity of the algorithm. Params represents the total number of parameters that need to be trained in the model. Test time/samples is the test time of each test sample. The smaller the value of FLOPs, Params, and Test time/samples, the higher the computational efficiency of the model.

## B. Comparison With Other Methods

For the comparative experiment, we consider six models. They are ARIMA [48], ConvLSTM [42], PredRNN [46], MIM [50], STConvS2S-C [44], and STConvS2S-R [44]. The ARIMA [48] model is a traditional statistical method for time series prediction. The ConvLSTM [42] architecture is a classic model for spatiotemporal sequence prediction. PredRNN [46] and MIM [50] models are currently more advanced spatiotemporal sequence prediction algorithms. STConvS2S-C [44] and STConvS2S-R [44] frameworks are currently more advanced models that only use convolutional layers for weather forecasting. The SA-JSTN model is inspired by the PredRNN framework; therefore, PredRNN can be employed as the baseline of the proposed architecture. The same time pattern and space coverage are adopted for different models. In experiments, the number of stacking layers ( $L$ ) is four, the size of filters ( $K$ ) is  $5 \times 5$ , and the number of hidden states ( $H$ ) is 64 for all models. To avoid overfitting in the training process, dropout is applied after the convolutional layer and set to 0.5. The SA-JSTN model applies the early stopping method on the validation dataset and sets the number of early stopping iterations to 50000. We have trained and evaluated each model ten times and, then, calculated MAE and RMSE on the test set. The experimental results are shown in Table I.  $T$  means the total input steps.  $T'$  means the total forecasting steps. The results MAE and RMSE in Table I are the average results over  $T'$  steps.

From the results, the SA-JSTN model is much better than other models on MAE and RMSE. The performance of the SA-JSTN model is better than that of the ARIMA model, which indicates that adding spatial dimension modeling on the basis of temporal dimension modeling can effectively capture the spatiotemporal characteristics of temperature images. The performance of the SA-JSTN model is much better than that of the ConvLSTM model, which illustrates the importance of modeling

TABLE II  
EVALUATION OF DIFFERENT SETTINGS FOR PREDRNN AND SA-JSTN

Dataset	Horizon	Version	Settings	PredRNN				SA-JSTN			
				MAE	RMSE	FLOPs	Params	MAE	RMSE	FLOPs	Params
CFSR	$T = 5, T' = 5$	1	L=2,K=3,H=32	1.2005	1.7014	<b>1.22G</b>	<b>0.23M</b>	0.8745	1.2652	1.23G	0.24M
		2	L=3,K=3,H=32	1.1904	1.6875	1.98G	0.38M	0.8633	1.2548	2.01G	0.39M
		3	L=4,K=3,H=64	1.1844	1.6887	10.97G	2.14M	0.8597	1.2308	11.07G	2.16M
		4	L=4,K=5,H=64	1.1736	1.6859	30.13G	5.88M	<b>0.8422</b>	<b>1.2273</b>	30.24G	5.91M

The boldface values represent the best results of the five evaluation indicators MAE, RMSE, FLOPs, Params, and Test time/samples on different data sets.

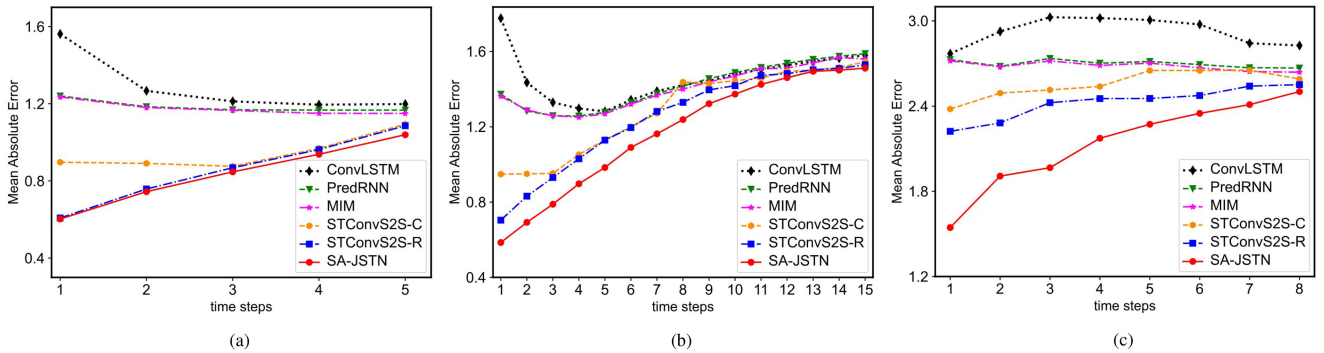


Fig. 6. Frame-wise MAE comparisons of different models. (a) CFSR test set ( $T' = 5$ ). (b) CFSR test set ( $T' = 15$ ). (c) HMBC test set ( $T' = 8$ ).

the spatiotemporal dimensions in a unified unit and using the self-attention module to capture the correlation of the spatial dimension. The SA-JSTN model is better than PredRNN and MIM, which illustrates that the self-attention module can capture the global spatial context information of the temperature image well. The STConvS2S-C and STConvS2S-R models only exploit the sequence modeling architecture of the 3-D convolution layer to learn the correlation of spatial and temporal in weather data. The SA-JSTN model is also superior to the STConvS2S-C and STConvS2S-R models, which indicates that LSTM has stronger long-term memory capture ability than CNN. From the results of Table I, the FLOPs, Params, and Test time/samples of the ConvLSTM model are the smallest. The MIM framework has the lowest computational efficiency. The FLOPs, Params, and Test time/samples of the SA-JSTN architecture are comparable to that of PredRNN, and slightly lower than those of STConvS2S-C and STConvS2S-R. The SA-JSTN model has achieved a reduction in prediction errors on both the CFSR dataset and the HMBC dataset. The SA-JSTN framework has achieved an improvement in prediction accuracy without a significant increase in FLOPs, Params, and Test time/samples.

### C. Analysis and Discussion

To study the best hyperparameters suitable for SA-JSTN model and prove its insensitivity to parameters, we adopt different the number of stacking layers, the size of filters, and the number of hidden states to compare the baseline model PredRNN with SA-JSTN. In the first prediction strategy of the

CFSR dataset, we set four versions of the parameters, as shown in Table II. In the case of using MAE and RMSE as evaluation metrics, we discover that when the number of stacked layers (L) is four, the size of filters (K) is  $5 \times 5$ , and the number of hidden states (H) is 64, the model obtains the best results. The SA-JSTN architecture is better than that of the PredRNN model in all four versions. In the comparative experiment of the previous section, we also adopted the same hyperparameters as the fourth version. Another noteworthy point is that increasing the size of filters, the number of hidden states, and the number of stacked layers can improve the predicted results to some extent. But overall, the performances of the four versions are basically stable. It indicates that the proposed model is not sensitive to parameters. In the case of using FLOPs and Params as evaluation metrics, we discover that as the size of the filters, the number of hidden states, and the number of stacked layers increase, the computational efficiency of PredRNN and SA-JSTN decreases. It is worth noting that the size of the filters and the number of hidden states have a greater impact on the FLOPs and Params of the model.

The frame-by-frame quantitative comparison of different models is shown in Figs. 6 and 7. A lower value indicates better prediction performance. The SA-JSTN model is always better than other models. In the first prediction strategy of the CFSR dataset, the SA-JSTN model is significantly better than the ConvLSTM, PredRNN, and MIM, and its results are equivalent to STConvS2S-C and STConvS2S-R architectures. In the second prediction strategy of the CFSR dataset, the SA-JSTN model is more accurate for short-term prediction, and the prediction results of the last few frames are equivalent to STConvS2S-C and



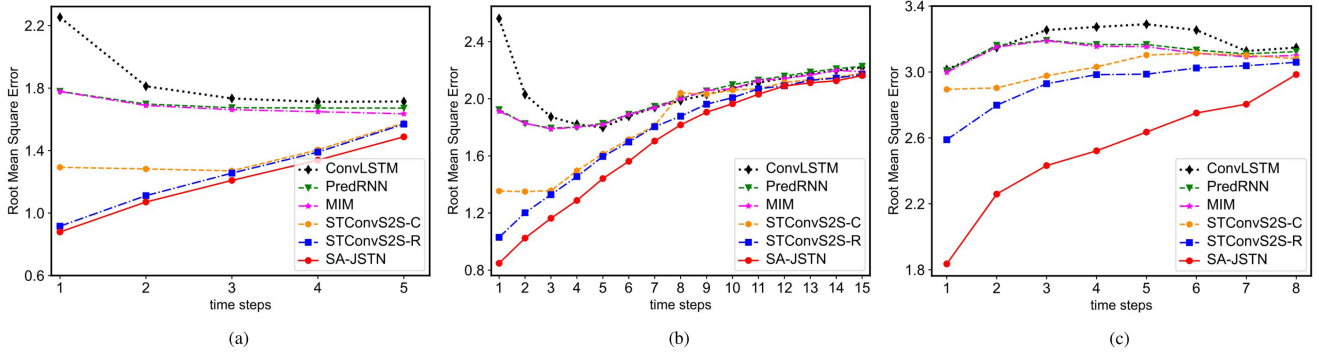


Fig. 7. Frame-wise RMSE comparisons of different models. (a) CFSR test set ( $T' = 5$ ). (b) CFSR test set ( $T' = 15$ ). (c) HMBC test set ( $T' = 8$ ).

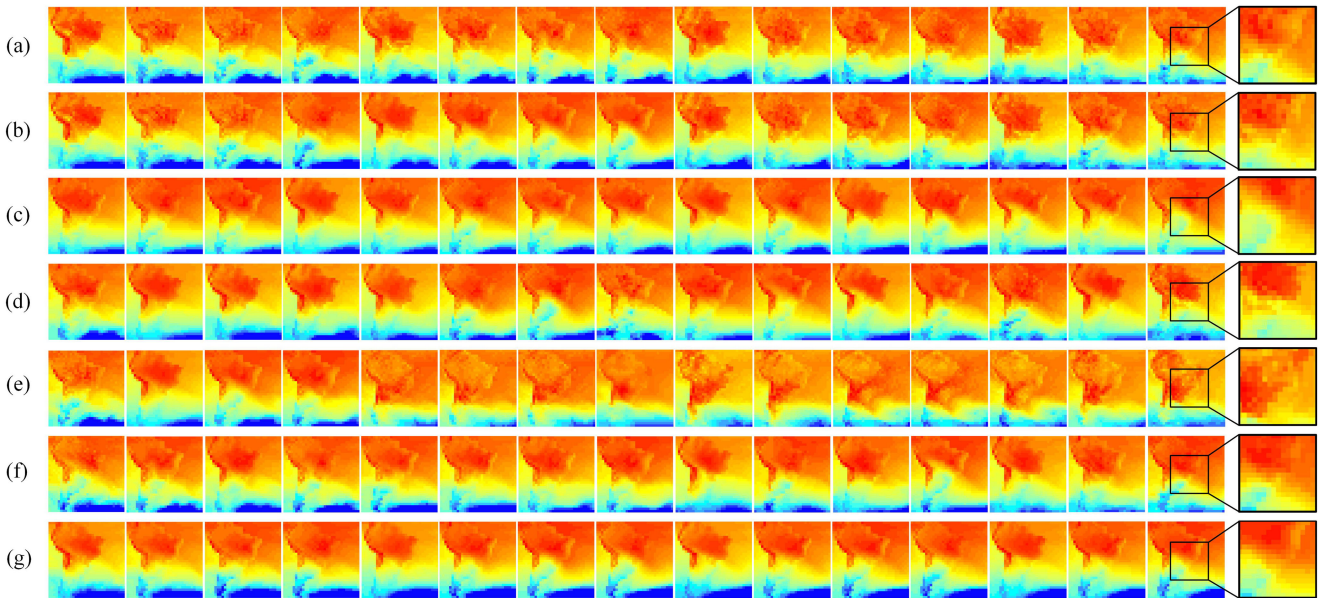


Fig. 8. Prediction examples on the CFSR test set ( $T' = 15$ ). We magnify the local of prediction results for additional detailed comparison at the last frame. (a) Ground truth. (b) SA-JSTN. (c) ConvLSTM. (d) PredRNN. (e) MIM. (f) STConvS2S-C. (g) STConvS2S-R.

STConvS2S-R architectures. In the HMBC dataset, the MAE and RMSE indicators of the SA-JSTN model are obviously lower than those of other network architectures. Due to the accumulation of errors, the prediction performance gradually decreases with the increase of the number of predicted frames. Overall, our model has a better effect for short-term temperature prediction.

In practice, the temperature change in the sample area of the HMBC test set with a pixel size of  $100 \times 100$  ( $H \times W$ ) is not obvious, and the visual effect is poor. To better compare the qualitative results of different models, we choose the CFSR test set with more obvious visual effects for display. Fig. 8 shows a temperature sample label sequence in the second strategy of CFSR test set and the prediction results of different models. To compare the prediction effectiveness of different models on spatiotemporal data in detail, we enlarged the local area of the last frame. The result of ConvLSTM network is obviously inaccurate because it cannot remember the detailed spatial representation. PredRNN and MIM models are more accurate than ConvLSTM, but their ability to predict some details is poor. The

STConvS2S-C and STConvS2S-R frameworks produce more accurate prediction results. However, the STConvS2S-C and STConvS2S-R models lack the LSTM layer to model long-term motion, which leads to the effectiveness of STConvS2S-C and STConvS2S-R becoming worse as time increases. SA-JSTN generates clearer prediction results and can memorize detailed visual appearance and long-term movements.

## V. CONCLUSION

In this article, we propose an end-to-end network, called SA-JSTN, which is applied for spatiotemporal sequence temperature forecasting learning and captures the temperature change process in both spatial and temporal dimensions. In addition, we present an STM unit, which adopts a dual-memory gating structure as the kernel component of SA-JSTN. The STM module ensures that the memory states capture the abstract spatial characteristics in the horizontal direction as well as running through all the time states in the vertical direction. Experiments show that the SA-JSTN model achieves the most advanced performance on

two atmospheric temperature datasets, and can better analyze the spatial and temporal correlations in the temperature prediction. Therefore, when utilizing spatiotemporal data for temperature prediction, SA-JSTN may be a natural choice for spatiotemporal sequence modeling tasks (such as weather forecasting). The SA-JSTN model performs well in the short-term temperature forecasting process and has achieved competitive results for long-term prediction. Future work will look for ways to reduce errors in long-term temperature forecasting and improve calculation efficiency. In addition, we will explore more spatial and temporal prediction domain architectures.

#### ACKNOWLEDGMENT

The authors would like to thank Rafaela Castro of the Federal Center for Technological Education of Rio de Janeiro, Brazil, and the Hebei Meteorological Bureau of China, for providing the CFSR and HMBC datasets in the study, respectively.

#### REFERENCES

- [1] H. Li, S. Gao, G. Liu, D. Guo, C. Grecos, and P. Ren, "Visual prediction of typhoon clouds with hierarchical generative adversarial networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 9, pp. 1478–1482, Sep. 2020.
- [2] N. He, L. Fang, and A. Plaza, "Hybrid first and second order attention Unet for building segmentation in remote sensing images," *Sc. China Inf. Sci.*, vol. 63, no. 4, 2020, Art. no. 140305.
- [3] W. Li, Z. Zou, and Z. Shi, "Deep matting for cloud detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8490–8502, Dec. 2020.
- [4] T. Haritini *et al.*, "Health impact assessment for mortality associated with high temperatures in Cyprus," in *Proc. 18th Mediterranean Electrotechnical Conf.*, 2016, pp. 1–4.
- [5] L. Lin and F. Weng, "Estimation of hurricane maximum wind speed using temperature anomaly derived from advanced technology microwave sounder," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 5, pp. 639–643, May 2018.
- [6] C. D. Rodgers, "Retrieval of atmospheric temperature and composition from remote measurements of thermal radiation," *Rev. Geophys.*, vol. 14, no. 4, pp. 609–624, 1976.
- [7] M. D. Goldberg, Y. Qu, L. M. McMillin, W. Wolf, L. Zhou, and M. Divakarla, "AIRS near-real-time products and algorithms in support of operational numerical weather prediction," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 2, pp. 379–389, Feb. 2003.
- [8] W. J. Blackwell, "A neural-network technique for the retrieval of atmospheric temperature and moisture profiles from high spectral resolution sounding data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 11, pp. 2535–2546, Nov. 2005.
- [9] J. L. Case, F. J. LaFontaine, J. R. Bell, G. J. Jedlovec, S. V. Kumar, and C. D. Peters-Lidard, "A real-time MODIS vegetation product for land surface and numerical weather prediction models," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 3, pp. 1772–1786, Mar. 2014.
- [10] C. Surussavadee, "Evaluation of high-resolution tropical weather forecasts using satellite passive millimeter-wave observations," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2780–2787, May 2014.
- [11] M. Huang, B. Huang, and H.-L. A. Huang, "Acceleration of the WRF Monin–Obukhov–Janjic surface layer parameterization scheme on a MIC-based platform for weather forecast," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 10, pp. 4399–4408, Oct. 2017.
- [12] X. Zhang, J. Zhou, F.-M. Göttsche, W. Zhan, S. Liu, and R. Cao, "A method based on temporal component decomposition for estimating 1-km all-weather land surface temperature by merging satellite thermal infrared and passive microwave observations," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4670–4691, Jul. 2019.
- [13] X. Ouyang, D. Chen, and Y. Lei, "A generalized evaluation scheme for comparing temperature products from satellite observations, numerical weather model, and ground measurements over the Tibetan plateau," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 7, pp. 3876–3894, Jul. 2018.
- [14] H. Li, H. Liu, M. Duan, X. Deng, and S. Zhang, "Estimation of air temperature under cloudy conditions using satellite-based cloud products," *IEEE Geosci. Remote Sens. Lett.*, 2021.
- [15] T. A. Chawsheen and M. Broom, "Seasonal time-series modeling and forecasting of monthly mean temperature for decision making in the Kurdistan region of Iraq," *J. Stat. Theory Pract.*, vol. 11, no. 4, pp. 604–633, 2017.
- [16] S. Curceac, C. Ternynck, T. B. M. J. Ouarda, F. Chebana, and S. D. Niang, "Short-term air temperature forecasting using nonparametric functional data analysis and SARMA models," *Environ. Model. Softw.*, vol. 111, pp. 394–408, 2019.
- [17] M. A. Jallal, S. Chabaa, A. El Yassini, A. Zeroual, and S. Ibyaich, "Air temperature forecasting using artificial neural networks with delayed exogenous input," in *Proc. IEEE Int. Conf. Wirel. Technol., Embedded Intell. Syst.*, 2019, pp. 1–6.
- [18] J. Y. Kajuru, K. Abdulkarim, and M. M. Muhammed, "Forecasting performance of ARIMA and SARIMA models on monthly average temperature of Zaria, Nigeria," *ATBU J. Sci., Technol. Educ.*, vol. 7, no. 3, pp. 205–212, 2019.
- [19] T. S. Skiksi and L. S. Al-Blooshi, "Climate change in the UAE: Modeling air temperature using ARIMA and STI across four bio-climatic zones," *F1000 Res.*, vol. 8, no. 973, 2019, Art. no. 973.
- [20] D. T. Meshram, S. D. Gorantiwar, N. Bake, and S. S. Wadne, "Forecasting of air temperature of western part of Maharashtra, India," *Int. J. Sci. Environ. Technol.*, vol. 8, no. 1, pp. 201–217, 2019.
- [21] H. Wang, J. Huang, H. Zhou, L. Zhao, and Y. Yuan, "An integrated variational mode decomposition and ARIMA model to forecast air temperature," *Sustainability*, vol. 11, no. 15, 2019, Art. no. 4018.
- [22] Y. Lai and D. A. Dzombak, "Use of the autoregressive integrated moving average (ARIMA) model to forecast near-term regional temperature and precipitation," *Weather Forecasting*, vol. 35, no. 3, pp. 959–976, 2020.
- [23] W. Wanishakpong and B. E. Owusu, "Optimal time series model for forecasting monthly temperature in the southwestern region of Thailand," *Model. Earth Syst. Environ.*, vol. 6, no. 1, pp. 525–532, 2020.
- [24] M. Farsi *et al.*, "Parallel genetic algorithms for optimizing the SARIMA model for better forecasting of the NCDC weather data," *Alexandria Eng. J.*, vol. 60, no. 1, pp. 1299–1316, 2021.
- [25] A. K. Mandal, D. Mallick, R. Sah, A. Ali, and M. A. Habib, "Artificial neural network models for temperature forecasting in Sundarban region of Bangladesh," *J. Sci. Technol.*, vol. 81, pp. 81–88, 2019.
- [26] J. A. Lazzús, P. Vega-Jorquera, I. Salfate, F. Cuturrufo, and L. Palma-Chilla, "Variability and forecasting of air temperature in Elqui Valley (Chile)," *Earth Sci. Inform.*, vol. 13, no. 4, pp. 1411–1425, 2020.
- [27] G. K. Rahul, S. Singh, and S. Dubey, "Weather forecasting using artificial neural networks," in *Proc. IEEE 8th Int. Conf. Rel., Infocom. Technol. Optim. (Trends Future Directions)*, 2020, pp. 21–26.
- [28] T. T. K. Tran, S. M. Bateni, S. J. Ki, and H. Vosoughifar, "A review of neural networks for air temperature forecasting," *Water*, vol. 13, no. 9, 2021, Art. no. 1294.
- [29] X. Zhang, Y. Li, S. Gao, and P. Ren, "Ocean wave height series prediction with numerical long short-term memory," *J. Mar. Sci. Eng.*, vol. 9, no. 5, 2021, Art. no. 514.
- [30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [31] M. A. Zaytar and C. El Amrani, "Sequence to sequence weather forecasting with long short-term memory recurrent neural networks," *Int. J. Comput. Appl.*, vol. 143, no. 11, pp. 7–11, 2016.
- [32] Q. Zhang, H. Wang, J. Dong, G. Zhong, and X. Sun, "Prediction of sea surface temperature using long short-term memory," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1745–1749, Oct. 2017.
- [33] J. Liu, T. Zhang, G. Han, and Y. Gou, "TD-LSTM: temporal dependence-based LSTM networks for marine temperature prediction," *Sensors*, vol. 18, no. 11, 2018, Art. no. 3797.
- [34] A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta, "Single layer & multi-layer long short-term memory (LSTM) model with intermediate variables for weather forecasting," *Procedia Comput. Sci.*, vol. 135, pp. 89–98, 2018.
- [35] C. Li, Y. Zhang, and G. Zhao, "Deep learning with long short-term memory networks for air temperature predictions," in *Proc. IEEE Int. Conf. Artif. Intell. Adv. Manuf.*, 2019, pp. 243–249.
- [36] X. Liu, T. Wilson, P.-N. Tan, and L. Luo, "Hierarchical LSTM framework for long-term sea surface temperature forecasting," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics*, 2019, pp. 41–50.



- [37] C. Xiao, N. Chen, C. Hu, K. Wang, J. Gong, and Z. Chen, "Short and mid-term sea surface temperature prediction using time-series satellite data and LSTM-AdaBoost combination approach," *Remote Sens. Environ.*, vol. 233, 2019, Art. no. 111358.
- [38] J. L. Charco, T. Roque-Colt, K. Egas-Arizala, C. M. Pérez-Espinoza, and A. Cruz-Chóez, "Using multivariate time series data via long-short term memory network for temperature forecasting," in *Proc. Int. Conf. Syst. Inf. Sci.*, 2020, pp. 38–47.
- [39] M. M. R. Khan, M. A. B. Siddique, S. Sakib, A. Aziz, I. K. Tasawar, and Z. Hossain, "Prediction of temperature and rainfall in Bangladesh using long short term memory recurrent neural networks," in *Proc. IEEE 4th Int. Symp. Multidisciplinary Stud. Innov. Technol.*, 2020, pp. 1–6.
- [40] T. Wu, C. Liu, and C. He, "Prediction of regional temperature change trend based on LSTM algorithm," in *Proc. IEEE 4th Int. Technol., Netw., Electron. Automat. Control Conf.*, 2020, vol. 1, pp. 62–66.
- [41] P. Hewage, M. Trovati, E. Pereira, and A. Behera, "Deep learning-based effective fine-grained weather forecasting model," *Pattern Anal. Appl.*, vol. 24, no. 1, pp. 343–366, 2021.
- [42] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 802–810.
- [43] Z. Karevan and J. A. Suykens, "Spatio-temporal stacked LSTM for temperature prediction in weather forecasting," 2018.
- [44] R. Castro, Y. M. Souto, E. Ogasawara, F. Porto, and E. Bezerra, "STConvS2S: Spatiotemporal convolutional sequence to sequence network for weather forecasting," *Neurocomputing*, vol. 426, pp. 285–298, 2020.
- [45] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.
- [46] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, "PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 879–888.
- [47] Y. Yang, J. Dong, X. Sun, E. Lima, Q. Mu, and X. Wang, "A CFCC-LSTM model for sea surface temperature prediction," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 207–211, Feb. 2018.
- [48] C. N. Babu and B. E. Reddy, "Predictive data mining on average global temperature using variants of ARIMA models," in *Proc. IEEE-Int. Conf. Adv. Eng. Sci. Manage.*, 2012, pp. 256–260.
- [49] 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. 5123–5132.
- [50] 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/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9146–9154.
- [51] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [52] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 451–466.
- [53] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5669–5678.



**Lukui Shi** received the B.S. degree in computer and application and the M.S. degree in computer application technology from the Hebei University of Technology, Tianjin, China, in 1996 and 1999, respectively, and the Ph.D. degree in computer application technology from Tianjin University, Tianjin, China, in 2006.

Since 2014, he has been a Professor with the School of Artificial Intelligence, Hebei University of Technology. He has authored two books and more than 20 articles. His research interests include machine

learning, lung sound recognition, and data digging.

Dr. Shi was a Member of the Discrete Intelligent Computing Professional Committee of the Chinese Association for Artificial Intelligence, and the Visual Big Data Professional Committee of China Society of Image and Graphics.



**Nanying Liang** received the B.S. degree in computer science and technology from Shanxi Datong University, Datong, China, in 2019. She is currently working toward the M.S. degree in software engineering with the School of Artificial Intelligence, Hebei University of Technology, Tianjin, China.

Her research interests include machine learning and intelligent computing.



**Xia Xu** (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in control science and engineering, from the School of Electrical Engineering, Yanshan University, Qinhuangdao, China, in 2012 and 2015, respectively, and the Ph.D. degree in pattern recognition and intelligent system from the School of Astronautics, Beihang University, Beijing, China, in 2019.

She is currently an Assistant Professor with the College of Computer Science, Nankai University, Tianjin, China. Her research interests include hy-

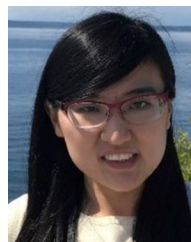
perspectral unmixing, multiobjective optimization, and remote sensing image processing.



**Tao Li** (Member, IEEE) received the Ph.D. degree in computer science from Nankai University, Tianjin, China, in 2007.

He is currently a professor with the College of Computer Science, Nankai University. His research interests include heterogeneous computing, machine learning, and Internet of Things.

Dr. Li is a Member of the IEEE Computer Society and ACM, and a Distinguished Member of the CCF.



**Zhou Zhang** received the B.S. degree in astronautics engineering and the M.S. degree in instrumentation science and optoelectronics engineering from Beihang University, Beijing, China, in 2010 and 2013, respectively, and the Ph.D. degree in geomatics and civil engineering from Purdue University, West Lafayette, IN, USA, in 2017.

She is currently an Assistant Professor with the Department of Biological Systems Engineering, University of Wisconsin-Madison, Madison, WI, USA. Her research interests include multisource data fusion,

UAV-based imaging platforms developments, hyperspectral image analysis, geospatial data analysis, and machine learning and its related applications to remote sensing data.