

Fine-Grained Conditional Convolution Network With Geographic Features for Temperature Prediction

Cheng Zhang, Guoshuai Zhao^{1b}, *Member, IEEE*, Junjiao Liu, Xingjun Zhang, *Member, IEEE*, and Xueming Qian^{2b}, *Member, IEEE*

Abstract—Short-to-medium term temperature prediction in high resolution is a very challenging task, involving meteorology, physics, mathematics, geography, and many other subjects. Its purpose is to fit a complex function from historical meteorological data to predict the future 1–5 days temperature, which is a typical spatio-temporal prediction problem. Meteorological data show complex correlations in local space. Most of the existing machine learning methods are based on image pixel-level tasks or spatio-temporal prediction tasks, which model meteorological data without considering the characteristics of meteorological data and use rough global patterns to model local space which would lose many details. To address the above issues, our work fine-grained conditional convolution network (FCCN) proposes a novel grid-level conditional convolution module, including a local geographic adaptive weight (GAW) and a local data adaptive weight (DAW). These two components are integrated into a multiscale meteorological fusion gated recurrent unit (GRU) architecture for the end-to-end temperature prediction. Experiments in real-world datasets from ERA-5 show our FCCN model has a better performance than all other baseline methods.

Index Terms—Conditional convolution, deep neural network, multiscale feature, temperature prediction.

I. INTRODUCTION

WEATHER prediction is a work that uses past weather data to predict future weather, such as temperature and wind. It is closely related to our life. Precise prediction of weather can affect the decision of industrial and agricultural

Manuscript received 6 January 2023; revised 26 April 2023; accepted 30 June 2023. Date of publication 24 July 2023; date of current version 9 August 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 61902309, Grant 62272380, and Grant 62103317; in part by the China Postdoctoral Science Foundation under Grant 2020M683496 and Grant BX20190273; in part by the Fundamental Research Funds for the Central Universities, China, under Grant xxj022019003 and Grant xzd012022006; and in part by the Science and Technology Program of Xi'an, China, under Grant 21RGZN0017. (Corresponding author: Guoshuai Zhao.)

Cheng Zhang, Guoshuai Zhao, and Junjiao Liu are with the School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China, and also with Shaanxi Yulan Jiuzhou Intelligent Optoelectronic Technology Company Ltd., Xi'an 710049, China (e-mail: zc981001@stu.xjtu.edu.cn; guoshuai.zhao@xjtu.edu.cn; liujunjiao@stu.xjtu.edu.cn).

Xingjun Zhang is with the School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: xjzhang@mail.xjtu.edu.cn).

Xueming Qian is with the Ministry of Education Key Laboratory for Intelligent Networks and Network Security and the SMILES Laboratory, School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: qianxm@mail.xjtu.edu.cn).

Digital Object Identifier 10.1109/TGRS.2023.3298318

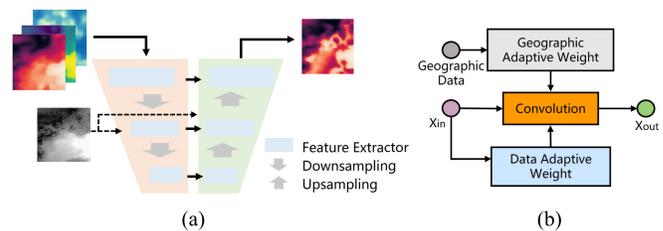


Fig. 1. Overview of our proposed model. (a) Multiscale meteorological fusion architecture. (b) Fine-grained conditional convolution. The historical weather data are modeled by the multiscale feature extractor and each feature extractor is composed of ConvGRU with fine-grained conditional convolution.

production [1]. Meteorological data have a strong spatio-temporal correlation, and sufficient modeling of the correlation between time and space dimensions is the key to weather prediction.

In the traditional numerical model numerical weather prediction (NWP [2], [3]), the meteorological data are calculated by a large number of mathematical and physical expressions to obtain a fine-grained spatio-temporal prediction, but this often requires the supercomputer. Machine learning algorithms, such as autoregressive integrated moving average (ARIMA [4], [5]), usually regard weather prediction problems as time series and ignore the complex spatial relationships which are very important for this task. With the development of deep learning, pixel-level vision tasks like semantic segmentation are used to deal with grid-based weather data. In addition, ConvRNN [6], which is based on spatio-temporal series prediction, is also widely used in weather prediction. Most of the existing deep learning models have a good performance on whole-world weather data, but not on fine-grained local data with less than 1° resolution. In our work, our attention is paid to fine-grained local temperature forecasting task.

Pixel-level vision tasks [7], [8], [9] usually use different-scale convolution models to decode on the backbone to get pixel-level output. There is not such a large dataset similar to ImageNet [10] in meteorological forecast tasks, so we cannot train a general backbone such as ResNet [11]. We need to directly model meteorological data. Spatial modeling of meteorological data usually uses image modeling strategies such as convolution [12], [13], or vision transformer [14], [15], but the geographical characteristics of meteorological data are not well captured by this kind of model. The model usually

performs well with simple geographical distributions such as sea surface and plain, but it performs poorly in other complex geographical distributions. In addition, the intervals of points in meteorological data are not strictly sampled equally. As the latitude increases, the distance between adjacent points with equal longitude is shrinking. Meanwhile, meteorology presents a strong local geographic relevance, i.e., the closer the two points are, the more similar the climate pattern is. In conclusion, the main challenge can be summarized as follows: 1) the geographical relations are not fully utilized for weather modeling, which is important in NWP and 2) the existing models do not sufficiently use the meteorological prior knowledge. How to use meteorological knowledge to organize the models is a difficult point to be explored.

In this work, we propose a temperature prediction model fine-grained conditional convolution network (FCCN), which can effectively combine geographical and meteorological prior knowledge into weather modeling. The main contributions of this article can be summarized as follows.

- 1) We propose a dynamic convolution mechanism in which convolution kernel parameters are shown in Fig. 1(b), which consist of a local geographic adaptive part, a local data adaptive part, and a global shared part. The dynamic convolution mechanism can effectively add context features to the model and enhance the local modeling capability and reduce the prediction error of different climate patterns.
- 2) We construct a multiscale meteorological fusion architecture as shown in Fig. 1(a). We use the dynamic convolution operator mentioned above and the multiscale semantic information extraction module to build the FCCN. Same to the multiscale meteorological theory in meteorology, the information with different scales in our model for weather prediction is useful.
- 3) The method we proposed was tested on the data of Beijing, Xi'an, and London in the ERA5 real weather dataset, and achieves state-of-the-art (SOTA) performance in the local temperature prediction task.

II. RELATED WORK

The temperature prediction problem is usually regarded as a spatio-temporal prediction problem with grid-level granularity, and different scholars have different insights into this problem. This section will be developed in the following aspects: 1) we introduce image-based meteorological modeling methods, then we introduce spatial-temporal-based meteorological modeling methods and 2) finally, we introduce conditional convolution and its application in weather prediction.

A. Image-Based Meteorological Modeling

Some scholars model the meteorological data as a multichannel image. They concatenate the feature dimension and temporal dimension of the meteorological data and treat temperature prediction as a transformation task from one multichannel image to another multichannel image [16], [17], [18], [19], [20], and [21]. This task is similar to the image semantic segmentation task in computer vision, and could

reuse some deep learning models of pixel-level tasks. Rasp and Thuerey [22] use 19 layers ResNet to predict global mesoscale weather at 5.625° resolution, and the u/v wind and temperature are predicted. This work models the whole world as an image with 32×64 size, and the meteorological information of four-time slices is stacked to predict the weather of the next three and five days. It uses the ERA-5 and CMIP-6 datasets and uses two strategies (the direct approach and the continuous approach) to make the prediction. FourCastNet [23] uses the adaptive Fourier neural operators (AFNO) [24] (an improved Vi-T model) to model global 0.25×0.25 meteorological data as a 720×1440 high-resolution image. A continuous approach "Fine-Tuning" is used for training, i.e., a slice X_k , which stands for the weather data of the k th time, is used to predict the next slice X_{k+1} , and then the X_{k+1} is used to predict the X_{k+2} , and so on iteratively. For each step of the prediction, the model is updated by backpropagation. Among the above models, FourCastNet uses the smallest resolution of 0.25° (about 55 km at the equator), which cannot use for real-life production directly. In addition, the above models are evaluated globally, while the ocean area accounts for 0.71 of the total area, the weather patterns of the ocean are simple, and the landscape is largely undulating, which reduces the difficulty of temperature prediction.

B. Spatio-Temporal-Based Meteorological Modeling

Other scholars have built meteorological prediction models based on spatio-temporal prediction problems [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], mostly based on ConvRNN [6] model with slight improvements. These models are basically a mixture of ConvRNN and full convolutional network, and are completely based on the data-driven training method for meteorological prediction. MetNet [36] uses ConvLSTM as the temporal encoder and axial attention [37] as the spatial aggregator to model the spatial and temporal features of meteorological data serially. The model treats the precipitation problem as a multiple regression task and has some improvement compared to the high-resolution rapid refresh model (HRRR) baseline. Deep generative models of radar (DGMR) [38] uses the convolutional network as the encoder and Conv gated recurrent unit (GRU) as the decoder for precipitation prediction and uses the generative adversarial strategy to ensure the spatio-temporal stability of the prediction generated by the generator. This model has been evaluated by experts to achieve better prediction results and produce greater economic benefits. Ms-Nowcasting [39] uses ConvLSTM as the encoder of meteorological data to predict precipitation, and proposes to use the output of HRRR as part of the input to add the physical operator to the model. It provides an idea that using a large area of $1280 \times 1280 \text{ km}^2$ to predict future precipitation data for a central local area of $256 \times 256 \text{ km}^2$. ConvRNN can capture spatial correlation while modeling temporal series and can greatly reduce computational complexity. Using ConvRNN as the backbone is the best choice for spatio-temporal modeling and is widely used in the processing of meteorological data.

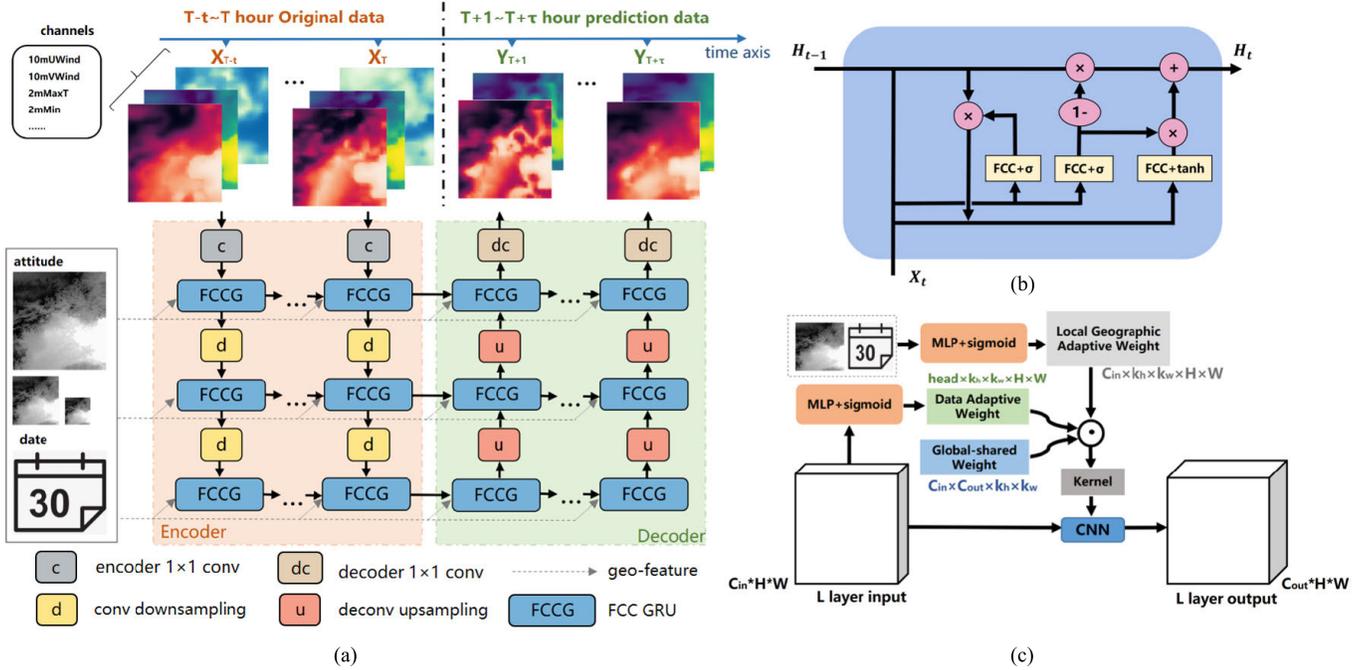


Fig. 2. Graph of FCC network framework. (a) Overall architecture. c is a convolution operation with kernel size of 1×1 , and stride of 1 for a projection from input dimension to embedding dimension; d is a convolution operation with kernel size of 2×2 , padding of 0, and stride of 2; u is a deconvolution operation with a kernel size of 4×4 , padding of 1 and stride of 2; dc is convolution operation with kernel size of 1×1 , stride of 1 for a projection to output dimension. (b) FCCG cell architecture. (c) Fine-grained conditional convolution operator architecture.

C. Conditional Convolution in CV and Meteorological Modeling

Convolution is often used for weather prediction due to its efficiency in processing grid point data [40]. The fine-grained weather data can also be modeled by convolution because of the same data format, but it has some differences. The propagation patterns of weather data are affected by geography, season, and specific weather conditions. The vanilla convolution kernel models a feature map with the same pattern. Therefore, a uniform global shared convolution kernel does not capture these local patterns effectively. CoAtNet [41] proposes to increase the receptive field of CNN by adding the convolutional kernel with adaptive attention weight, and the model achieves the SOTA results on ImageNet. CondConv [42] parameterizes the convolution kernel with a linear combination of multiple expert knowledge, and the expert weights are obtained by using GAP + FC + Sigmoid on the input feature map of the current layer. ODConv [43] leverages a novel multidimensional attention mechanism to compute four types of attention for the convolution kernel along all four dimensions of the kernel space in a parallel manner. These works [44], [45], and [46] have improved the convolution kernel by adding dynamic weights to the global-shared convolution kernel on sample-by-sample and layer-by-layer level. Conditional convolution increases the size and capacity of the network, improves the global modeling capability of the model, and enhances the performance of the model. However, these conditional convolution methods are not fully applicable to weather prediction due to the unique local correlation of weather data. Conditional local convolution (CLC) [47] for graph-based weather prediction proposes to use

local conditional kernels for message aggregation instead of the original graph convolutional kernels, and the effectiveness of the idea is experimentally demonstrated.

Referring to the previous experience and the problems existing in the model, we introduce the dynamic convolution network into meteorological modeling to find more weather patterns for accurate temperature prediction. Compared with the other dynamic convolution networks, our method has large differences in parameter generation method, motivation, and application scenario. Our model establishes a grid point-level dynamic convolution operator and inserts the operator into a multiscale ConvGRU.

III. METHODOLOGY

In this section, we first give the mathematical definition of the grid-based spatio-temporal weather prediction problem we are to solve in this article. Then, we give a complete description of our model. Finally, we focus on our proposed convolution operator, fine-gain conditional convolution (FCC).

A. Problem Definition

In the grid-based spatio-temporal prediction problem, the t th frame of the input data can be represented as $X_t \in R^{H*W*C}$, where H and W represent the height and the weight of the input data, and C represents the channel size, like a dense image matrix. T frames are concatenated in the time dimension to obtain the overall spatio-temporal data $X \in R^{H*W*C*T}$. There are also some contextual features G like date and altitude, which is useful in this task. Our problem is to fit a function F based on the last data and contextual features

to make a downscaled prediction of the future weather Y . The mapping function can be expressed as follows:

$$\{X_{T-t}, \dots, X_{T-1}, X_T, G\} \xrightarrow{F} \{Y_{T+1}, \dots, Y_{T+\tau}\} \quad (1)$$

where $X_i \in R^{H*W*C_{in}}$ is meteorological data in i th time slice, $Y_i \in R^{H*W*C_{out}}$ is temperature data in i th time slice, $G \in R^{H*W*C_G}$ is contextual feature data in i th time slice, C_{in}, C_{out}, C_G represent the channel size of the input data, the output temperature data and the contextual feature data, and F is a regression function with learnable parameters.

B. Overview of the Proposed Network

We present the framework of FCCN shown in Fig. 2(a). It consists of a stacked spatio-temporal encoder and decoder layer. As shown in Fig. 2, a spatio-temporal encoder is constructed by convolution block, fine-grained conditional convolution GRU (FCCG), and downsampling block. A decoder is constructed by deconvolution block, FCCG, and upsampling block. FCCG will be introduced in the next paragraph. By stacking multiple spatio-temporal layers, FCCN is able to handle spatial dependencies at different temporal levels. When predicting $T + 1 \sim T + \tau$ hour temperature, we need $T - t \sim T$ hour original weather data and contextual feature data. The original weather data first be encoded into feature maps of three sizes by a c block, FCCG, d block, FCCG, d block, FCCG, where c, d stand for a convolution block, downsampling block. Then the decoder will use the three feature maps to get the prediction data by FCCG, u , FCCG, u , FCCG, and dc , where the u and dc are upsampling block and convolution block. The contextual feature includes attitude and date data. The attitude data is processed by the max-pooling operator into three sizes corresponding to the feature map. The date data are embedded and broadcasted to different sizes like feature maps. The operation details of the contextual feature will be discussed in Section III-C.

We choose MSE as the loss function of FCCN. Similar to the large viewport (LV) prediction method in MS-nowcasting [39], we also use the method that the input viewport is larger than the target. The input tensor of our model is $C_{in} \times 64 \times 64$, and the predicted tensor is $C_{out} \times 32 \times 32$, which is cropped from the central input region geographically. The loss function can be expressed as follows:

$$\text{loss}(X_n, Y_n) = \sum_{i=1}^{32} \sum_{j=1}^{32} (X_{n,i+16,j+16} - Y_{n,i,j})^2 \quad (2)$$

where $X_n \in R^{64*64*C}$, $Y_n \in R^{32*32*C}$ is cropped from the input central area and upper left point offset of horizontal and vertical coordinates is eight, standing for the n th ground truth and prediction, C is the number of the output channel.

C. Multiscale FCCG Architecture

The ConvRNN model replaces the fully connected networks with the convolution networks in RNN, which can model 2-D time series data like meteorological data. GRU is a variant of RNN that uses reset gate and update gate to avoid gradient

vanishing problems and extract features better in time series data. Referring to the recent algorithms, ConvGRU is popular in weather forecast tasks, and [47] experiment on 1-D-CNN, RNN, and GRU for meteorological data modeling, which shows the effectiveness of GRU. In addition, image-based models ignore the correlation of meteorological data in time series, and we use GRU to model the time dimension to effectively capture the characteristics of spatio-temporal series. To sum up, we use the fine-grained conditional convolution operator to replace the convolution operator in ConvGRU for the weather information encoder and decoder, named FCCG. Both the encoder and decoder of FCCN have three FCCG layers, and the specific settings of each layer will be discussed in the next paragraph. The encoder part is similar to the regular sequence prediction task. In the encoder stage, the output of each hidden layer at time t is downsampled twice as the input of the next hidden layer. The bottom layer of the decoder uses the full zero tensor as the input, and the hidden state output of the encoder is used as the hidden state input of the decoder. To match the encoder hidden state output, each hidden layer output of the decoder is upsampled two times as the input of the next layer. We set the output of the decoder's last layer as the result of the temperature prediction. The mathematical formulation of FCCG can be defined as

$$z_t = \sigma(fcc_z(\text{concat}(x_t|h_{t-1}))) \quad (3)$$

$$r_t = \sigma(fcc_r(\text{concat}(x_t|h_{t-1}))) \quad (4)$$

$$\tilde{h}_t = \tanh(fcc_h(\text{concat}(x_t|r_t \circ h_{t-1}))) \quad (5)$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t \quad (6)$$

where h_{t-1}, z_t, r_t, h_t is the last hidden state, the update gate, the reset gate, and the hidden state, respectively, fcc is a fine-grained conditional convolution operator, \circ represents the Hadamard product.

Fusion features from different scales can increase the receptive field of the model and capture the hierarchical features of meteorological data. In the traditional weather forecast workflow, for a system in the atmosphere, we can determine the scale of the system according to the spatial range, simulate the atmospheric diffusion of different scales, and then aggregate forecast information from different scales. On the spatial scale, the atmospheric motion can be divided into several scales. For example, the range of the atmospheric boundary layer is 1 km; the range of the convection cell and the cumulonimbus is 10 km; the range of the frontal surface and the squall line is 100 km. So, using the information of different scales can simulate atmosphere changes better and predict future weather better. We use convolution layers whose convolution kernel size is 3×3 and stride is 2 for downsampling between each layer, and we use three-layer FCCG to get three different scale features. Unlike the skip connection operation of U-net, a GRU block accepts both input and hidden state data. We use the different scale features that the output in the previous step as the hidden state input to each decoder, and this operation can replace the feature concatenation operation. On the decoder, upsampling is performed between each layer using a convolution operation

with the convolution kernel size of 4×4 , the stride of 2, and the padding of 1.

D. Fine-Grained Conditional Convolution

The meteorological pattern should have the following properties [47]: location-characterized, smooth, and common. Location-characterized emphasizes that local convolution kernels should be similar to local meteorological patterns. smooth emphasizes the smoothness of the convolution kernels, i.e., points that are physically close or have similar local spaces should have similar convolution kernels, and common emphasizes that the local convolution kernels are adjusted according to the local space but shared across different local spaces. Based on the inspiration of these three points, we design fine-grained conditional convolution.

1) *Global Shared Weight*: Unlike the conditional convolution with global shared dynamic parameters, our dynamic parameters are specific to each operation in the convolution window. The parameters of the convolution kernel of each grid point are from three parts, the global shared convolution kernel parameters, the local geographic data adaptive convolution parameters, and the local data adaptive convolution parameters. The global shared weight (GSW) w is like the vanilla convolutional kernel, which is shared by all raster points and is responsible for aggregating information from different regions. The role of the GSW is to add different weights to the information aggregated from each direction. Due to the different local conditions, the information aggregation weights are different from each direction, and the weight of global shared convolutional kernel is trained by all points.

2) *Geographic Adaptive Weight*: The local geographic adaptive weight (GAW) is related to the local meteorological pattern, which is determined by the local geographic feature and season together. For the convolution central grid point i , its geographic feature and month information can be expressed as m_{i0} , and its eight surrounding neighbor grid points are m_{ij} , $1 \leq j \leq 8$. We assume that its local weather pattern kernel is $\phi(m_{i0})$, and the local convolution kernel $\phi(mi0)$ is given by $\{m_{i0}|m_{ij}, 1 \leq j \leq 8\}$, we use multilayer perceptron to approximate this local GAW and use the sigmoid activation function to confine the data to between 0 and 1, as

$$\phi(m_{i0}) = \text{Sigmoid}(\text{mlp}(m_{i0}|m_{ij})), \quad 1 \leq j \leq 8 \quad (7)$$

where mlp is a two-layer linear model with a *leakyrelu* activation function between the layers and $i \in [h \times w]$, denoting all the raster points of a frame. The output channel of $\phi(m_{i0})$ is equal to the input channel of the layer.

3) *Data Adaptive Weight*: The last part of the convolution kernel is calculated by the value of local meteorological data, which is called the local data adaptive weight (DAW). The specific value of local meteorological data also affects the aggregation of the local information. For grid point i , the specific value of meteorological data is x_{i0} , and its neighbor grid points are $\{x_{ij}, 1 \leq j \leq 8\}$, which determines the convolution kernel of meteorological elements $\psi(xi)$, we use another multilayer perceptron to fit the local meteorological DAW and use sigmoid to limit the values of the parameters.

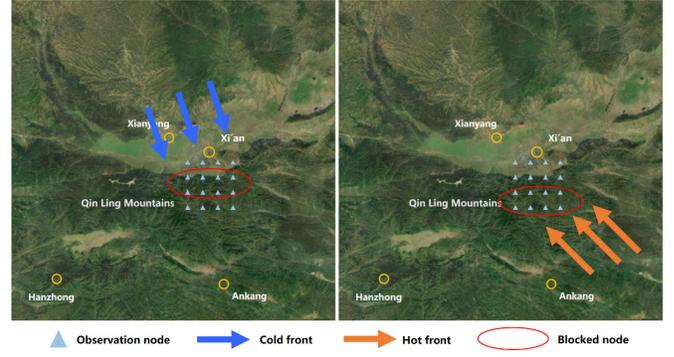


Fig. 3. Schematic of atmospheric propagation in Xi'an.

The multihead self-attention structure used in the transformer has a good result on machine translation tasks. The multiheads can capture multiple dependencies between tokens. So, we use the multihead mechanism to capture more information patterns in the adaptive convolution kernel of meteorological data. The local DAW is as follows:

$$\psi_k(x_{i0}) = \text{Sigmoid}(\text{mlp}_k(x_{i0}|x_{ij})), \quad 1 \leq j \leq 8 \quad (8)$$

$$\psi(x_{i0}) = \text{concat}(\psi_k(x_{i0})), \quad 1 \leq k \leq K \quad (9)$$

where i and j are defined in the same way as in the previous local GAW, k denotes the k th heads and K is the number of the heads. K mlp operations produce the K heads of the data adaptive convolution kernel, and concatenating the K heads produces the final data adaptive convolution kernel. The output channel of $\psi(x_{i0})$ is equal to the output channel of this layer. If we obtain the parameters of the three components, the overall fine-grained convolution operation is expressed mathematically as follows:

$$f_{cc}(x_{in}) = [\phi(m_i) \odot w \odot \psi(x_i)] * x_{in} + b \quad (10)$$

where $*$ is convolution operation, w , $\phi(m_i)$, and $\psi(x_i)$ represent GSW, local GAW, and local DAW, respectively. b is the bias which is like vanilla convolution.

Local geographical features have a very important impact on the propagation of the atmosphere. Different landforms have different blocking effects on meteorological transmission. In addition, atmospheric pressure, light, and other factors also affect meteorological propagation. As shown in Fig. 3, Xi'an is affected by the cold front from the north in winter and Qin Ling Mountains blocks this spread. So, the north of the Qinling Mountains presents a dry and cold weather pattern, while the Qinling Mountains and the south are less affected by the cold front; When the warm front from the south flows northward, the south of the Qinling Mountains presents a humid and hot weather pattern, and the Qinling Mountains and the north are less disturbed. The difference in meteorological conditions and terrain has caused strong irregular propagation to the atmosphere. Our algorithm dynamically adjusts the convolution kernel parameters according to the terrain and weather data, so that the appropriate strategy of local information aggregation is used. Compared with the static convolution operation, our model has a better performance in irregular weather

patterns. Our method FCC can adaptively adjust convolution parameters based on data and geographical features, explicitly modeling the laws of atmospheric propagation, and capturing local meteorological patterns more effectively than static parameter convolution. Therefore, FCC is more effective in local meteorological modeling.

IV. EXPERIMENT

This section introduces the experiments in detail. Here, Section IV-A the details of our dataset are introduced, Section IV-B presents baselines, metrics, and details of our experiment, Section IV-C reports the performance measurements, Section IV-D discusses root mean square error (RMSE) curves of five prediction days and season-by-season comparison, and Section IV-E provides a discussion on ablation study and parameter sensitivity.

A. Dataset

We build our experiments on the ERA-5 dataset [48], which is the fifth generation European Centre for Medium-Range Weather Forecasts (ECMWF) atmospheric reanalysis of the global climate. Reanalysis combines model data with observations from across the world into a globally complete and consistent dataset. As shown in Table I, We select data from three regions, covering London area ($-2.7^{\circ}\sim 3.7^{\circ}\text{N}$, $41.3^{\circ}\sim 53.7^{\circ}\text{E}$), Beijing area ($36.8^{\circ}\sim 43.2^{\circ}\text{N}$, $113.3^{\circ}\sim 119.7^{\circ}\text{E}$), and Xi'an area ($30.4^{\circ}\sim 36.7^{\circ}\text{N}$, $105.3^{\circ}\sim 111.6^{\circ}\text{E}$), with an input grid of 64×64 points, with a horizontal resolution of $0.1^{\circ} \times 0.1^{\circ}$, and an output grid of 32×32 . We choose 22 factors broadly based on meteorological intuition as used in model output machine learning (MOLR) [49]. The data are from January 1, 2002 to December 31, 2018, and the data slice interval is 6 h, and one day is split into four intervals. In addition, the elevation of each region in China was obtained from the PTPE dataset, and the elevation of London area was obtained from Google Maps interface. The altitude statistics of three areas can be found in Table II. We selected January 1, 2002, to December 31, 2016, as the training set, including 5478 days and 21 912 frames of meteorogram, used 365 days of data from January 1, 2017, to December 31, 2017, as the evaluation set, and used 365 days of data from January 1, 2018, to December 31, 2018, as the test set for all three regions.

B. Baselines, Metrics, and Details

We used the following five baselines to compare the performance with FCCN.

- 1) **MOLR** [49] uses linear regression for weather prediction, and we use year-around and adjacency point feature engineering methods for processing, concatenating the elevation, and date information with the original data, and establish a three-layer linear regression.
- 2) **RN19** [22] uses a deep residual convolutional neural network to predict wind, temperature, and precipitation.
- 3) **FourCastNet** [23] is a vision transformer-based weather prediction model, which replaces the transformer operator with an ANFO.

- 4) **UU-Net** [50] regards weather prediction as image segmentation and use U-net to capture multiscale features.
- 5) **Ms-Net** [39] uses three layers ConvLSTM for weather prediction, we follow the architecture of the model and use MS-nowcasting+LV.

In models which do not use time series modeling, the time dimension is simply merged into the feature channel dimension. And we use 64–32 LV operations to all models. The evaluation metrics for the experimental results are RMSE and mean absolute error (MAE). RMSE and MAE are the common evaluation metrics for regression tasks.

Our models and all deep learning baselines are implemented by PyTorch. All methods are evaluated on a Linux server with two GPUs. We sample all initialization values from normal distribution, and use Adam [51] as the optimizer. The initial learning rate is set to $1e-3$. We reduce the learning rate at Plateau with factor = 0.25 and practice = 5 for learning rate adjustment, and early stopping is set to 20, which means the learning rate is reduced to 0.25 times of the last one when the loss function does not decrease with five epoch at evaluation set. All models use the same input variables, the lead time is 24 h with four-time slices, and the output is the maximum and minimum temperature for the next five days. The code will be released at <https://github.com/DraymonKey/FCCN>.

C. Performance Comparison

Table III shows the overall performance of the five baselines and our model. We report the five-day average RMSE and average MAE. Underlined are the best results for the baselines, and bolded are the best results for all models. Our model achieves the best results for all but the maximum temperature of London RMSE, and the difference between our model and the best baseline is small. Specifically, our model achieves (5.3%, 7.3%) and (5.2%, 6.6%) improvements beyond the best baseline on MAE and RMSE on the Beijing dataset for (maximum and minimum temperature) prediction, respectively. Similarly, the improvement of MAE and RMSE on Xi'an dataset is (6.4%, 6.4%) and (5.7%, 5.5%), respectively. This indicates that our model has the best performance compared to baselines.

Different area prediction results vary greatly because of geographic factors. The climate of the Beijing area is the warm temperate humid monsoon, with large temperature variations and the largest forecast errors in our experiment. Xi'an areas have a higher average elevation and gentler undulations than Beijing, showing a trend of high, low, and high-and-low from northwest to southeast, and the prediction error is smaller than in Beijing. In the results of Table III, expect the RMSE of the maximum temperature close to MS-Net, we still show an improvement in other prediction indicators. But the performances of each method in London are very similar and good. From Table II, we can find that 94.53% of the grid points in London are below 200 m above sea level, and the other grid points are below 400 m. Meanwhile, London is a typical plain terrain and presents a marine climate, and the climate pattern is simple and far more predictable. We use

TABLE I
STATISTICS OF DATASET

Area	Input Coordinate	Point Numbers	Number of Frames	Train Dataset Time Span	Train Dataset Size	Valid Dataset Size	Test Dataset Size
Beijing	36.8° ~ 43.2°N, 113.3° ~ 119.7°E	1024	21912	2002.1.1~2016.12.31	5478	365	365
Xi'an	30.4° ~ 36.7°N, 105.3° ~ 111.6°E						
London	-2.7° ~ 3.7°N, 41.3° ~ 53.7°E						

TABLE II
ALTITUDE STATISTICS OF THREE AREAS

Altitude [m]	≤ 200	200~400	400~600	600~800	800~1000	>1000
Beijing	568	48	85	98	71	154
Xi'an	0	66	164	150	141	503
London	968	56	0	0	0	0

a dynamic convolution kernel to extract more local weather patterns, but the performance in the plain has not been greatly improved. However, it is worth noting that our model performs better in complex terrain than others. We also show an instance of maximum temperature prediction result and RMSE results for each quarter in Appendix.

Different models have different prediction results due to different modeling capabilities. In all models, MOLR and RN19 have the worst results, which we believe is caused by less consideration of information at different scales and temporal information. FCN is a class of Vi-T-based models with large representational capacity, and it is easily overfitting in our dataset leading to poor generalization performance. In addition, the UU-Net model fuses the features of different scales and shows a great improvement over RN19 which only considers only one scale. This proves the effectiveness of modeling at different scales for weather prediction. The MS-Net model, which is a spatio-temporal prediction model, further enhances the effectiveness of only spatial prediction models. Our model simulates local meteorological models and performs spatio-temporal prediction at multiscale.

D. Five Days Result and Different Quarters Result

We also report the RMSE of each model for the future five days temperature prediction results, as shown in Fig. 4. From Fig. 4, it can be seen that MS-Net and ours with dynamic decoder perform better than others, which indicates that temporal feature and spatial feature are equally important for the temperature prediction problem. RN19 and MOLR models are less effective than other models, and it is not effective to use scale-invariant network to fit the transfer of meteorological information in complex terrain. In addition, the FCN, which has a larger model capacity, has comparable performance to the MS-Net around two days forecast time, but is less effective than MS-Net for 3–5 days of prediction. Also, UU-Net, which performs multiscale feature fusion, has reduced performance after the first three days. MS-Net, which considers spatio-temporal features, has good performance in 1–5 days. It is noteworthy that our model has the best performance from one to five days, and still maintains a good performance as time goes on. The multiscale information fusion increases the

perceptual field on the one hand and captures hierarchical weather features at different levels on the other.

In the results of the Xi'an region, similar to the prediction results for the Beijing region, the model with only spatial modeling has good results on the first day, and there is a sharp performance decline as the prediction time moves on. MS-Net, which has the ability of spatio-temporal modeling, has a better effect result within five days. The result of our model in one day is similar to the MS-net's, but ours has a better effect over 2–5 days. This is due to the fact that conditional convolution can simulate meteorological patterns well, which is similar to local meteorological patterns. The meteorological information would be stopped by the Qinling Mountains, but would be well propagated in the Guanzhong Plain, which we will discuss in Section IV-E. Overall, our model has good performance in the long time series compared to other models.

E. Ablation Study and Parameter Sensitivity

Then we study the effectiveness of each module in our model. The FCCN mainly includes fine-grained conditional convolution and multiscale feature fusion, and we will analyze these two parts separately. For the fine-grained conditional convolution, whose parameters come from three parts, the global shared convolution parameters are the same as the vanilla convolution, and we add two modules (geographic adaptive parameters and data adaptive parameters), respectively, for the ablation experiments on the vanilla convolution. To demonstrate the effectiveness of the two adaptive parameters, we also report and analyze Xi'an result finely. For the multiscale feature fusion module, we remove the downsampling block in the encoder and the upsampling block in the decoder for the ablation study.

The results of the ablation experiment in Beijing and Xi'an region can be found in Table IV. GSW, GAW, and DAW represent the GSW, GAW, and DAW. Both of our proposed convolution enhancement components improve the performance of vanilla convolution. When using vanilla convolution, our model gets only 2.9471 RMSE for the maximum temperature in Beijing, which is slightly higher than the performance of MS-Net. When only local GAWs are added to the model, the results are more significantly improved than vanilla convolution. Our model takes external factors such as topography and date information into full consideration, which improves the modeling ability of the model for complex terrain. It is worth noting that the addition of both volume modules to the general convolution has better results compared to the addition of a single module, which is because the two modules enhance the performance and capacity of the model in complex terrain and can capture more local patterns to improve the prediction of the model.

TABLE III
RESULT OF EXPERIMENT

Datasets	Predictor	Metrics	MOLR	FourCastNet	RN19	UU-Net	MS-Net	Ours
Beijing	max temperature	RMSE [K]	3.1033	3.0148	3.1233	3.0038	<u>2.9545</u>	2.7988
		MAE [K]	2.3212	2.2834	2.3484	2.2466	<u>2.2171</u>	2.0980
	min temperature	RMSE [K]	2.9509	2.8912	2.9635	2.8568	<u>2.8270</u>	2.6413
		MAE [K]	2.2027	2.1896	2.2298	2.1301	<u>2.1259</u>	1.9712
Xi'an	max temperature	RMSE [K]	2.9681	2.9829	2.9574	2.8981	<u>2.7919</u>	2.6327
		MAE [K]	2.2971	2.2891	2.2894	2.1959	<u>2.1397</u>	2.0028
	min temperature	RMSE [K]	2.7484	2.8580	2.7391	2.6717	<u>2.5848</u>	2.4427
		MAE [K]	2.1170	2.1814	2.1126	2.0237	<u>1.9764</u>	1.8491
London	max temperature	RMSE [K]	0.4702	0.4723	0.4668	0.4624	0.4499	0.4503
		MAE [K]	0.3405	0.3485	0.3392	0.3414	<u>0.3294</u>	0.3223
	min temperature	RMSE [K]	0.5026	0.5039	0.4997	0.4938	<u>0.4811</u>	0.4736
		MAE [K]	0.3590	0.3673	0.3575	0.3601	<u>0.3477</u>	0.3407

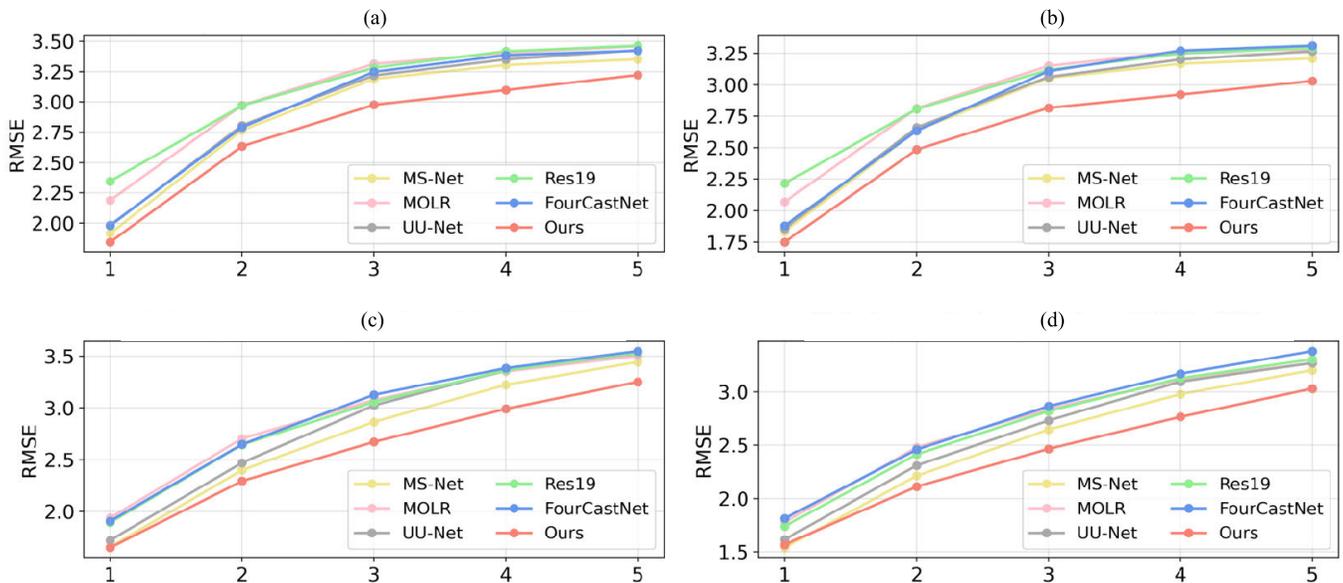


Fig. 4. We show five days RMSE result of Beijing and Xi'an. (a) and (b) Result of Beijing area. (c) and (d) Result of Xi'an area. Our model has the best result in the next five days around Beijing and Xi'an.

TABLE IV
ABLATION STUDY OF THE FINE-GRAINED CONDITIONAL CONVOLUTION

COMPONENTS			Beijing				Xi'an			
			max temperature		min temperature		max temperature		min temperature	
GSW	GAW	DAW	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
✓			2.9471	2.2052	2.8040	2.1023	2.784	2.1495	2.5603	1.9628
✓	✓		2.8359	2.1281	2.7048	2.0239	2.7341	2.0884	2.5067	1.8969
✓		✓	2.8988	2.1772	2.7529	2.0645	2.7011	2.0594	2.5376	1.9538
✓	✓	✓	2.7988	2.0980	2.6413	1.9712	2.6327	2.0028	2.4427	1.8491

We analyze the effectiveness of the multiscale feature fusion. In Table V, the “w/o u&d” means that the upsampling and downsampling blocks are not used in the model, so the feature map remains unchanged all the time. Compared with the model using uniform scale features, our model achieves an average RMSE improvement of 0.22 in Beijing and 0.15 in Xi'an. This proves the effectiveness of multiscale feature fusion. Fusion based on different scale features

can greatly improve the receptive field. In addition, the feature modeling of different scales in our model uses a strong prior knowledge, synoptic scale system, which proposes that different meteorological elements flow on different scales. The multiscale feature fusion effectively improves the modeling ability on climate patterns of different scales and improves the prediction effect of the model.

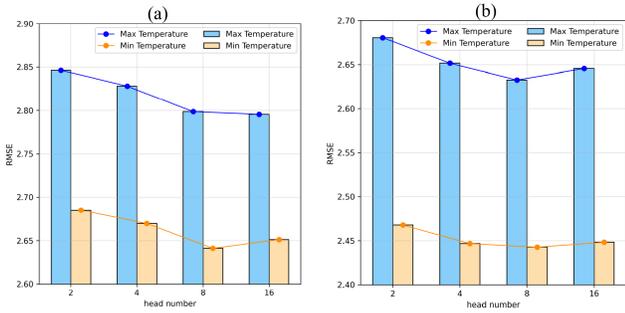


Fig. 5. Head number impact of the RMSE in Beijing and Xi'an area. (a) Impact of head numbers in Beijing. (b) Impact of head numbers in Xi'an.

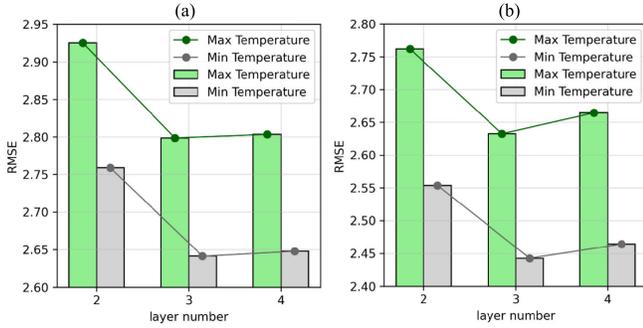


Fig. 6. Layer number impact of the RMSE in Beijing and Xi'an area. (a) Impact of layer number in Beijing. (b) Impact of layer number in Xi'an.

TABLE V

ABLATION STUDY OF THE UPSAMPLING AND DOWNSAMPLING BLOCKS

		Metrics	w/o u&d	ours
Beijing	max temperature	RMSE	3.0086	2.7988
		MAE	2.3633	2.098
	min temperature	RMSE	2.9845	2.6413
		MAE	2.2877	1.9712
Xi'an	max temperature	RMSE	2.8032	2.6327
		MAE	2.1416	2.0028
	min temperature	RMSE	2.5691	2.4427
		MAE	1.9614	1.8491

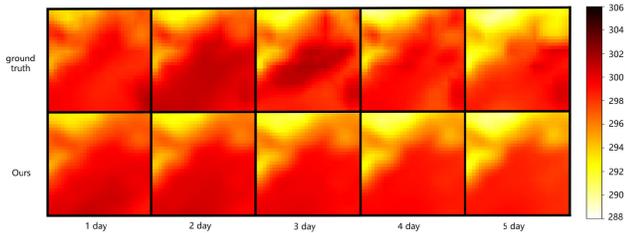


Fig. 7. We show 14 August~18 August 2018 maximum temperature prediction result of Beijing area.

Fig. 5 shows the impact of head numbers on the model. In Beijing area, when the number of heads is increased from 2 to 8, the performance of the model is improving. When the number of heads is increased from 8 to 16, the performance of the maximum temperature is slightly improved, and the minimum temperature result is the same as the eight heads. In addition, for the Xi'an area, when the number of heads increases from 8 to 16, the result of the model becomes bad. Fig. 6 shows the prediction results of the model layers number. When the number of layers increases from 2 to 3, the model effect is greatly improved. However, when the model layers

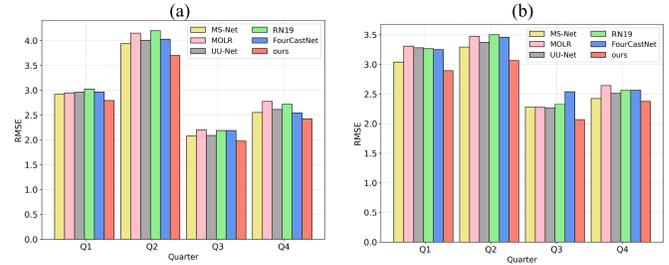


Fig. 8. We show four quarters max temperature RMSE result of Beijing and Xi'an. (a) Result of Beijing area. (b) Result of Xi'an area.

number is further enlarged, the model is more likely to be overfitting and the performance becomes worse. Our model works best when the number of heads is eight and the number of layers is three.

V. CONCLUSION

In this article, we propose a new fine-grained conditional convolution method to model local meteorological flows and predict temperature, which improves the capture ability of convolution methods for local meteorological patterns. We use two local adaptive parameter modules to constrain the convolution kernel so that the information aggregation strategy for a central point changes with the geographical location and meteorological feature. In addition, in order to better model meteorological data, we use the prior knowledge based on the synoptic scale system for multiscale feature fusion. Experiments have proved the effectiveness of our proposed method.

In the future, we think the incremental learning of meteorological data is necessary. Training models on large-scale weather data are very resource-intensive. But weather data are constantly being collected, and the latest data often carries some new weather trends and is important for prediction. How to learn new data with low resource consumption rates and restrain catastrophic forgetting is the future direction for weather prediction. Finally, we think a huge dataset, huge backbone, and training skills are also research focus, which is the same as the CV and NLP.

APPENDIX

MORE COMPARISON AND VISUALIZATION

As shown in Fig. 7, we obtained the maximum temperature prediction results of our model from August 14 to August 18 by using the data of August 13, 2018.

In Fig. 8, we show the RMSE of different quarters in the Beijing region and the Xi'an region. We divide the year into four quarters by month, January–March, April–June, July–September, and October–December. In the Beijing region, the prediction of the second quarter is particularly difficult and that of the third quarter is the easiest, with the RMSE D-value between these in our model being 2.7, and the first and fourth quarters RMSE falling in between. The temperature in the Beijing area will rise quickly in spring. Beijing's summer high temperature is strongly persistent and stable, with a small difference between day and night, so the model is easier to predict the temperature of the third quarter. Beijing's autumn is short. But the winter is cold and long. In Spring, the temperature of Xi'an rises quickly, but the rising is unstable,

making Q1 and Q2 forecasts more difficult. The summer in Xi'an is hot and rainy, and the temperature is stable in July and August, which is easy to predict. The fall season is influenced by the Pacific ironical maritime air and the winter in Xi'an is lack of precipitation, cold and dry. Like other deep learning models, the prediction result of our model is greatly affected by seasonal changes. In the second and third quarters, our model is more outstanding than the previous SOTA. In the first quarter and the fourth quarter, our model has little advantage.

REFERENCES

- [1] M. Reichstein et al., "Deep learning and process understanding for data-driven Earth system science," *Nature*, vol. 566, no. 7743, pp. 195–204, Feb. 2019.
- [2] P. Bauer, A. Thorpe, and G. Brunet, "The quiet revolution of numerical weather prediction," *Nature*, vol. 525, no. 7567, pp. 47–55, Sep. 2015.
- [3] D. J. Stensrud, M. C. Coniglio, K. H. Knopfmeier, and A. J. Clark, "Numerical models: Model physics parameterization," in *Encyclopedia of Atmospheric Sciences*. USA: Elsevier, 2015, pp. 167–180. [Online]. Available: <https://pure.psu.edu/en/publications/numerical-models-model-physics-parameterization>
- [4] E. De Saa and L. Ranathunga, "Comparison between ARIMA and deep learning models for temperature forecasting," 2020, *arXiv:2011.04452*.
- [5] R. D. Patil and O. S. Jadhav, "A big data prediction for weather forecast using hybrid ARIMA-ANN time series model," in *Recent Trends in Image Processing and Pattern Recognition*, K. C. Santosh and B. Gawali, Eds. Singapore: Springer, 2021, pp. 291–307.
- [6] 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*. Cambridge, MA, USA: MIT Press, 2015.
- [7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, *arXiv:1412.7062*.
- [8] 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.
- [9] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.
- [10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1–16.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 26th Annu. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.
- [13] Y. LeCun et al., "Handwritten digit recognition with a back-propagation network," in *Proc. Adv. Neural Inf. Process. Syst.* Burlington, MA, USA: Morgan Kaufmann, 1989, pp. 396–404.
- [14] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
- [15] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 9992–10002.
- [16] Q.-K. Tran and S.-K. Song, "Computer vision in precipitation nowcasting: Applying image quality assessment metrics for training deep neural networks," *Atmosphere*, vol. 10, no. 5, p. 244, May 2019.
- [17] M. Reza Ehsani, A. Zarei, H. V. Gupta, K. Barnard, and A. Behrangi, "Nowcasting-Nets: Deep neural network structures for precipitation nowcasting using IMERG," 2021, *arXiv:2108.06868*.
- [18] 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*.
- [19] J. A. Taylor, P. Larraondo, and B. R. de Supinski, "Data-driven global weather predictions at high resolutions," *Int. J. High Perform. Comput. Appl.*, vol. 36, no. 2, pp. 130–140, Mar. 2022.
- [20] A.-I. Albu, G. Czibula, A. Mihai, I. G. Czibula, S. Burcea, and A. Mezghani, "NeXtNow: A convolutional deep learning model for the prediction of weather radar data for nowcasting purposes," *Remote Sens.*, vol. 14, no. 16, p. 3890, Aug. 2022.
- [21] S. Moritani, T. Segal, S. Ishida, S. S. Mar, and B. A. O. Ahmed, "Regional climate fluctuation analysis using convolutional neural networks," *Earth Sci. Informat.*, vol. 15, no. 1, pp. 281–289, Mar. 2022.
- [22] S. Rasp and N. Thuerey, "Data-driven medium-range weather prediction with a resnet pretrained on climate simulations: A new model for WeatherBench," *J. Adv. Model. Earth Syst.*, vol. 13, no. 2, pp. 1–16, Feb. 2021.
- [23] J. Pathak et al., "FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators," 2022, *arXiv:2202.11214*.
- [24] J. Guibas, M. Mardani, Z. Li, A. Tao, A. Anandkumar, and B. Catanzaro, "Adaptive Fourier neural operators: Efficient token mixers for transformers," 2021, *arXiv:2111.13587*.
- [25] A. Kala and S. G. Vaidyanathan, "Prediction of rainfall using artificial neural network," in *Proc. Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Jul. 2018, pp. 339–342.
- [26] S. F. Tekin, O. Karaahmetoglu, F. Ilhan, I. Balaban, and S. S. Kozat, "Spatio-temporal weather forecasting and attention mechanism on convolutional LSTMs," 2021, *arXiv:2102.00696*.
- [27] L. Chen, Y. Cao, L. Ma, and J. Zhang, "A deep learning-based methodology for precipitation nowcasting with radar," *Earth Space Sci.*, vol. 7, no. 2, pp. 1–18, Feb. 2020.
- [28] L. Xu, N. Chen, Z. Chen, C. Zhang, and H. Yu, "Spatiotemporal forecasting in Earth system science: Methods, uncertainties, predictability and future directions," *Earth-Sci. Rev.*, vol. 222, Nov. 2021, Art. no. 103828.
- [29] Z. Yin et al., "Spatiotemporal fusion of land surface temperature based on a convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1808–1822, Feb. 2021.
- [30] Z. Ma, H. Zhang, and J. Liu, "PrecipLSTM: A meteorological spatiotemporal LSTM for precipitation nowcasting," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4109108.
- [31] Y. Wu, X. Yang, Y. Tang, C. Zhang, G. Zhang, and W. Zhang, "Inductive spatiotemporal graph convolutional networks for short-term quantitative precipitation forecasting," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5111618.
- [32] A. J. T. Sukanda and D. Adytia, "Wave forecast using bidirectional GRU and GRU method case study in Pangandaran, Indonesia," in *Proc. Int. Conf. Data Sci. Appl. (ICoDSA)*, 2022, pp. 278–282.
- [33] G. Cruz and A. Bernardino, "Learning temporal features for detection on maritime airborne video sequences using convolutional LSTM," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6565–6576, Sep. 2019.
- [34] H. V. Habi and H. Messer, "Recurrent neural network for rain estimation using commercial microwave links," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 3672–3681, May 2021.
- [35] Z. Karevan and J. A. K. Suykens, "Transductive LSTM for time-series prediction: An application to weather forecasting," *Neural Netw.*, vol. 125, pp. 1–9, May 2020.
- [36] C. Kaae Sønderby et al., "MetNet: A neural weather model for precipitation forecasting," 2020, *arXiv:2003.12140*.
- [37] J. Ho, N. Kalchbrenner, D. Weissborn, and T. Salimans, "Axial attention in multidimensional transformers," 2019, *arXiv:1912.12180*.
- [38] S. Ravuri et al., "Skillful precipitation nowcasting using deep generative models of radar," *Nature*, vol. 597, no. 9, pp. 672–677, 2021.
- [39] S. Kloccek et al., "MS-nowcasting: Operational precipitation nowcasting with convolutional LSTMs at Microsoft weather!" 2021, *arXiv:2111.09954*.
- [40] G. Ayzel, T. Scheffer, and M. Heistermann, "RainNet V1.0: A convolutional neural network for radar-based precipitation nowcasting," *Geosci. Model Develop.*, vol. 13, no. 6, pp. 2631–2644, Jun. 2020.
- [41] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "CoAtNet: Marrying convolution and attention for all data sizes," 2021, *arXiv:2106.04803*.
- [42] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "CondConv: Conditionally parameterized convolutions for efficient inference," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–14.
- [43] C. Li, A. Zhou, and A. Yao, "Omni-dimensional dynamic convolution," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–15.
- [44] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29. Red Hook, NY, USA: Curran Associates, 2016, pp. 1–16.
- [45] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, "Dynamic neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7436–7456, Nov. 2022.
- [46] S. A. Magid et al., "Dynamic high-pass filtering and multi-spectral attention for image super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4268–4277.

- [47] 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, vol. 36, no. 7, pp. 7470–7478.
- [48] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, and J. Thépaut, "The ERA5 global reanalysis," *Quart. J. Roy. Meteorolog. Soc.*, vol. 146, no. 730, pp. 1999–2049, 2020.
- [49] H. Li, C. Yu, J. Xia, Y. Wang, J. Zhu, and P. Zhang, "A model output machine learning method for grid temperature forecasts in the Beijing area," *Adv. Atmos. Sci.*, vol. 36, no. 10, pp. 1156–1170, Oct. 2019.
- [50] S. Choi, "UNet based future weather prediction on Weather4cast 2021 stage 2," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 5747–5749.
- [51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.



Cheng Zhang received the B.S. degree in software engineering from the Xi'an Jiaotong University of Technology, Xi'an, China, in 2016, where he is currently pursuing the M.S. degree in software engineering.

His research interests include Artificial Intelligence (AI) for science and online learning.



Guoshuai Zhao (Member, IEEE) received the B.E. degree from Heilongjiang University, Harbin, China, in 2012, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2015 and 2019, respectively.

He was an Intern with the Social Computing Group, Microsoft Research Asia, Beijing, China, from January 2017 to July 2017. He was a Visiting Scholar from October 2017 to October 2018 and with the Massachusetts Institute of Technology, Cambridge, MA, USA, from June 2019 to December

2019. He is currently an Associate Professor with Xi'an Jiaotong University. His research interests include social media big data analysis, recommender systems, and natural language generation.



Junjiao Liu received the B.S. degree from Lanzhou University, Lanzhou, China, in 2018. She is currently pursuing the M.S. degree in software engineering with Xi'an Jiaotong University, Xi'an, China.

Her research interests include AI for science and incremental learning.



Xingjun Zhang (Member, IEEE) received the Ph.D. degree in computer architecture from Xi'an Jiaotong University, Xi'an, China, in 2003.

From 1999 to 2005, he was a Lecturer and an Associate Professor with the Department of Computer Science and Technology, Xi'an Jiaotong University. From February 2006 to January 2009, he was a Research Fellow with the Department of Electronic Engineering, Aston University, Birmingham, U.K. From 2009 to 2013, he was an Associate Professor with the Department of

Computer Science and Engineering, Xi'an Jiaotong University, where he has been a Full Professor since 2014. His interests include high performance computer architecture, high performance computing, big data storage systems, and computer networks.



Xueming Qian (Member, IEEE) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree in electronics and information engineering from Xi'an Jiaotong University, Xi'an, in 2008.

From 2010 to 2011, he was a Visiting Scholar with Microsoft Research Asia, Beijing, China. He was previously an Assistant Professor with Xi'an Jiaotong University, where he was an Associate Professor from 2011 to 2014, and is currently a Full

Professor. He is also the Director of the Smiles Laboratory, Xi'an Jiaotong University. His research focuses on social media big data mining and search.