iTrendRNN: An Interpretable Trend-Aware RNN for Meteorological Spatiotemporal Prediction

Xu Huang[1], Chuyao Luo[2], Bowen Zhang[3*], Huiwei Lin[1], Xutao Li[1], Yunming Ye[1*]

[1]School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China
[2]Department of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen, China
[3]College of Big Data and Internet, Shenzhen Technology University, Shenzhen, China
huangxuu@outlook.com, luochuyao.dalian@gmail.com, zhang_bo_wen@foxmail.com, linhuiwei@stu.hit.edu.cn, lixutao@hit.edu.cn, yeyunming@hit.edu.cn

Abstract

Accurate prediction of meteorological elements, such as temperature and relative humidity, is important to human livelihood, early warning of extreme weather, and urban governance. Recently, neural network-based methods have shown impressive performance in this field. However, most of them are overcomplicated and impenetrable. In this paper, we propose a straightforward and interpretable differential framework, where the key lies in explicitly estimating the evolutionary trends. Specifically, three types of trends are exploited. (1) The proximity trend simply uses the most recent changes. It works well for approximately linear evolution. (2) The sequential trend explores the global information, aiming to capture the nonlinear dynamics. Here, we develop an attention-based trend unit to help memorize long-term features. (3) The flow trend is motivated by the nature of evolution, i.e., the heat or substance flows from one region to another. Here, we design a flow-aware attention unit. It can reflect the interactions via performing spatial attention over flow maps. Finally, we develop a trend fusion module to adaptively fuse the above three trends. Extensive experiments on two datasets demonstrate the effectiveness of our method.

Introduction

The spatiotemporal prediction of meteorological elements, such as temperature and relative humidity, is of great significance to our daily life. Conventional numerical weather prediction (NWP) methods mainly resort to specific mathematical models, which are known as computationally expensive and time-consuming. Later, data-driven shallow neural networks are utilized in this field. For example, Kuligowski et al. (Kuligowski and Barros 1998) employed a backpropagation neural network to predict 6-hour precipitation amounts.

In the last decade, deep neural networks have achieved remarkable success in many fields, including computer vision (Peng et al. 2022), natural language processing (Ma et al. 2023), etc. As for the spatiotemporal prediction tasks, existing methods can be grouped into two types. (1) The first one is based on convolutional neural networks (CNNs) (Zhang, Zheng, and Qi 2017; Xu et al. 2018; Gao et al. 2022b; Tan et al. 2023), where the spatial dependencies and temporal dynamics are both modeled by convolution operations. However, these methods show a weakness in capturing long-term sequential variations. (2) The second line also learns the spatial features by convolution, but employs recurrent neural networks (RNNs) to better capture the temporal dynamics (Shi et al. 2015; Wang et al. 2017, 2019; Lin et al. 2020; Wu et al. 2021; Huang et al. 2022b). Specifically, Shi et al. (Shi et al. 2015) first replaced the fully connected layer in long short-term memory (LSTM) units with convolution, and proposed a new model ConvLSTM. Afterwards, ConvLSTM has become one of the most representative benchmark in this filed. A variety of subsequent methods were developed to further improve the performance. For example, Wang et al. (Wang et al. 2017) introduced a spatiotemporal state to memorize both spatial appearances and temporal variations. Lin et al. (Lin et al. 2020) designed a self-attention mechanism to extract spatial features with both global and local dependencies.

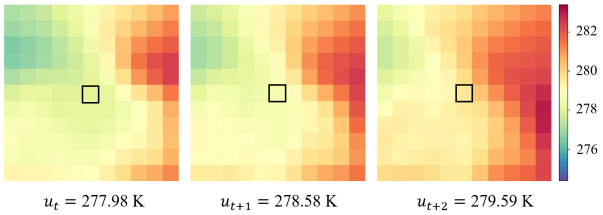


Figure 1: An example of temperature evolution. The values of central region (u) at time t, $t+1$, and $t+2$ are 277.98 K, 278.58 K, and 279.59 K, respectively. That is, it heats up by 0.6 K and 1.01 K. Most of the previous methods, such as PredRNN and TAU, directly predict u through implicit networks. In this paper, we explicitly estimate the change δ_u, which can better reflect the meteorological evolution.

Despite the effectiveness of previous studies, most of them rely on overcomplicated and obscure models. Researchers and users know little about their internals, thus limiting the development of this field. Recently, explainable artificial intelligence, which aims to reveal the internal mechanisms or decision basis of black-box models, has received increasing attention. However, existing studies mainly focused on the tasks of image classification (Zhou et al. 2016; Zhang et al. 2019, 2022) or time series anal-

*Both are Corresponding Authors.

ysis (Barić et al. 2021; Arras et al. 2019; Hou and Zhou 2020). As for the spatiotemporal prediction, there are few efforts devoted to the interpretability of models. Huang et al. (Huang et al. 2022a) first explored the internal mechanism of a modified ConvGRU (convolutional gated recurrent unit). Nevertheless, they only provided a post-hoc analysis, and the prediction model itself was not interpretable.

In this paper, we propose a self-explanatory framework for meteorological spatiotemporal prediction. As shown in Figure 1, the evolution of meteorological elements can be represented as a small increment based on their current values. Standing on this observation, we explicitly formulate the evolution by estimating the underlying trend. Specifically, as shown in Figure 2, three types of trends are exploited. (1) The proximity trend (PT) estimates the change at time t according to the change at time t-1. Obviously, PT works well when the curve is approximately linear at time t. (2) The second type is the sequential trend (ST). Instead of only using the proximity information, ST explores the long-term information. Compared with PT, ST performs better when the curve changes nonlinearly[1]. (3) The third type is the flow trend (FT). This is motivated by that the evolution is driven by the information interaction with surrounding regions. For example, as for the temperature, the heat tends to flow from hot to cold regions.

Based on the above three trends, we propose an interpretable trend-aware RNN, named iTrendRNN. The key of our model is a trend estimator, which has three modules for learning three trends and a fusion module for fusing them. Specifically, (1) the proximity trend is captured by a PT module, which directly replicates the change of the previous moment. (2) The sequential trend is modeled by an ST module. Here, we develop a new attention-based trend unit (ATU) to estimate the long-term trend. ATU can obtain better global information via performing temporal attention over historical features. Furthermore, we employ the true trend as a constraint to guide the learning process of ATU. (3) The flow trend is estimated by an FT module. Here, we propose a novel flow-aware attention unit (FAU). It performs spatial attention over flow maps, thereby reflecting the meteorological interactions. (4) Finally, we develop a trend attention mechanism to adaptively fuse these trends.

Overall, the main contributions of this paper are summarized as follows:

- We propose an interpretable trend-aware RNN (iTrendRNN) for meteorological spatiotemporal prediction. It follows a straightforward differential framework that explicitly formulates the evolution as an estimation of the trend.
- We develop three types of trends, i.e., proximity trend (PT), sequential trend (ST), and flow trend (FT). All of them can be well explained. Furthermore, we design a fusion module to adaptively fuse these trends.
- As for ST, we propose an attention-based trend unit (ATU) to better capture the long-term features. Moreover, its temporal weight can indicate the role of each time step. As for FT, we develop a flow-aware attention unit (FAU), where the meteorological interactions can be reflected by its spatial attention mechanism.
- We conduct extensive experiments to evaluate the proposed iTrendRNN. The results show that our model outperforms existing methods. We also perform a thorough analysis to investigate the role of different trends.

[1]In practice, the temperature within a day may usually rise first and then decrease. At the turning point, ST is the better choice.

Related Work

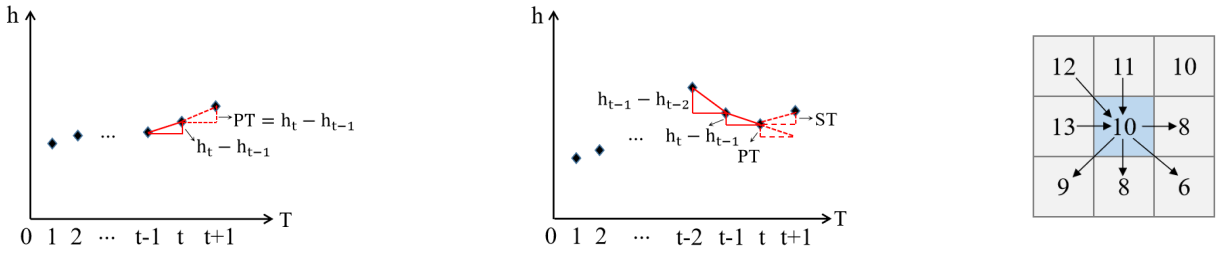
Spatiotemporal Prediction Models

The spatiotemporal prediction task aims to forecast future data frames based on historical observations. It covers many real-world applications, such as video prediction (Wu et al. 2021; Gao et al. 2022b), weather forecast (Shi et al. 2015, 2017), traffic flow prediction (Dai et al. 2022a; Xia, Jin, and Chen 2022), etc. In general, the prevailing studies can be divided into CNNs-based and ConvRNNs-based models. Specifically, (1) in the CNN-based line, convolution operations are used to capture not only spatial dependencies, but also temporal dynamics. For example, Zhang et al. (Zhang, Zheng, and Qi 2017) employed a residual CNN to model the spatiotemporal properties of crowd traffic. Gao et al. (Gao et al. 2022b) developed a CNN translator to learn temporal evolution. Nevertheless, most of them capture the dynamics via performing convolution in the time dimension, which can not effectively model the complex and long-term sequential changes (He, Chow, and Zhang 2020; Huang et al. 2023). (2) In the ConvRNN-based line, the temporal dynamics are handled by various RNNs. As a representative baseline, ConvLSTM (Shi et al. 2015) first integrated convolution into LSTM cells. Later, Wang et al. (Wang et al. 2017) proposed a spatiotemporal LSTM cell, which introduced a new state to memorize both spatial appearance and temporal variations. In (Wang et al. 2019), a memory-in-memory (MIM) block was designed to exploit the differential signals between adjacent states, aiming to model the non-stationary and approximately stationary dynamics. Recently, Lin et al. (Lin et al. 2020) proposed a novel self-attention ConvLSTM. They employed a self-attention memory module to capture features with long-range dependencies in terms of spatial and temporal domains. Moreover, generative adversarial network-based (Ravuri et al. 2021; Dai et al. 2022b) and transformer-based models (Gao et al. 2022a; Peng and Huang 2022) also attracted recent attention. However, the former is difficult to train, and the latter requires a lot of computing resources. Therefore, we do not discuss them in this work.

Although the above methods have shown impressive performance in specific tasks, they have become increasingly complex and difficult to understand. In this paper, we work towards a more transparent and interpretable model.

Explainable Artificial Intelligence

Explainable artificial intelligence aims to reveal the internal mechanisms or decision basis of black-box models. In terms of different networks for distinct tasks, existing studies can be grouped into three types. (1) The most widely studied is



(a) Proximity trend. (b) Sequential trend. (c) Flow trend.

Figure 2: Three types of trends in this work. In subfigures (a) and (b), the T-axis represents the time step, and the h-axis means the value of the hidden state. In subfigure (c), the arrows indicate the information flow into or out of the central region.

to explain CNNs for image classification tasks. For example, Zeiler et al. (Zeiler and Fergus 2014) first visualized the intermediate layers in CNNs via a deconvolutional network. The class activation mapping family of methods (Zhou et al. 2016; Selvaraju et al. 2017; Belharbi et al. 2022) provided the basis for classification by highlighting evidential image regions. Recently, Nauta et al. (Nauta et al. 2023) proposed to learn prototypical parts in a self-supervised fashion, which correlated better with human vision. (2) Explaining RNNs for time series analysis also attracted considerable interest. Herein, attention-based techniques are the most popular (Tran et al. 2018; Munkhdalai et al. 2019; Chien and Chen 2021; Papi, Negri, and Turchi 2022). They usually employed attention weights to reflect the importance of each time step, thus revealing the crucial parts. (3) Compared with CNNs and RNNs, there are few studies focusing on the interpretability of spatiotemporal prediction models. Huang et al. (Huang et al. 2023) proposed an interpretable local flow attention mechanism for traffic flow prediction. Beyond the similarity or correlation of features, Huang et al. (Huang et al. 2022a) suggested to explore the inner mechanism from two aspects, i.e., image generation analysis and spatiotemporal dynamics analysis.

In spite of the progress made by previous studies, an interpretable model for meteorological spatiotemporal prediction is under-researched. Particularly, the trend information, which well reflects the meteorological evolution, has never been explicitly exploited. In this paper, we propose an interpretable trend-aware model.

Methods

In this section, we first formally define the task, and then introduce our method in detail.

Task Definition

Meteorological Spatiotemporal Prediction Task (MSPT): Given a meteorological input sequence $u_{T-L+1:T} = \{u_{T-L+1}, ..., u_{T-1}, u_T\}$, MSPT aims to predict the most likely length-K sequence in the future, denoted as $\widehat{u}_{T+1:T+K} = \{\widehat{u}_{T+1}, ..., \widehat{u}_{T+K-1}, \widehat{u}_{T+K}\}$. Here, $u_t \in \mathbb{R}^{M \times N}$ is the observation at time t. L and K represent the lengths of the input and output sequences, respectively.

Motivation for Estimating Trends

To improve the interpretability of neural network-based methods, we propose to build a self-explanatory model from the perspective of estimating trends. Specifically, given a value $u(x, y, t_0 + \delta t)$, we have the following formula:

$$\begin{aligned} u(x, y, t_0 + \delta t) &= u(x, y, t_0) + (u(x, y, t_0 + \delta t) - u(x, y, t_0)) \\ &= u(x, y, t_0) + \delta u|_{t=t_0} \end{aligned} . \tag{1}$$

(x, y) indicates the position. t_0 means the current time. δt is the time increment, and δu is the corresponding increment of u, which is termed as trend in this work.

Eq. (1) effectively formulates the spatiotemporal evolution of meteorological elements. For example, the future temperature usually increases or decreases by a small increment based on the current value. In this work, we explicitly follow this equation to design our model, which thus can be easily understood and well explained.

Framework Overview

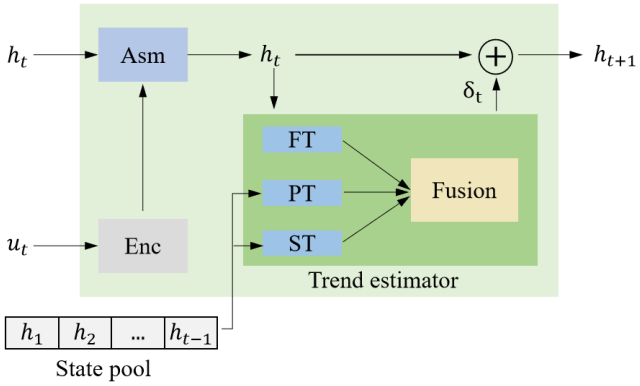


Figure 3: The structure of iTrendRNN. u_t is the input data, and h_t is the hidden state. Enc denotes the input encoder. Asm means the assimilation module. FT, PT, and ST are the flow trend, proximity trend, and sequential trend, respectively.

As described in the introduction part, three types of trends are exploited, namely proximity trend (PT), sequential trend (ST), and flow trend (FT). Based on them, we propose a new model iTrendRNN, which is shown in Figure 3. Here, to enhance the expressiveness, we estimate the trend on a hidden

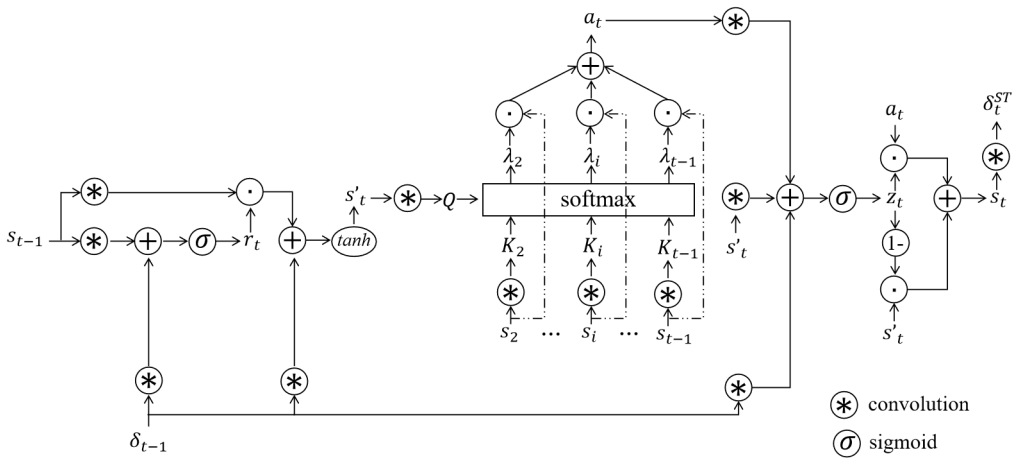


Figure 4: The inner structure of ATU. s memorizes the trend dynamics and δ_{t-1} is the most recent trend. Notice that s_1 and h_1 are initialized to 0, hence the temporal attention starts from $t = 2$.

space $\mathcal{H}$. Concretely, at time t, the input u_t is mapped by an encoder (Enc). Then we fuse the input data and hidden state ($h_t \in \mathcal{H}$) with an assimilation module (Asm)[2]. Asm can help h_t leverage the recent input data[3]. After that, h_t is sent to a trend estimator to obtain its trend δ_t. Finally, we update the hidden state as follows: $h_{t+1} = h_t + \delta_t$. h_{t+1} will be delivered to a CNN decoder to generate the prediction $\widehat{u}_{t+1}$.

In our framework, the evolution is explicitly formulated by $h_{t+1} = h_t + \delta_t$. This exactly follows Eq. (1). Next, we mainly introduce the trend estimator.

Trend Estimator

The trend estimator first calculates three types of trends, then utilizes a fusion module to fuse them.

Proximity Trend (PT) The idea of proximity trend (δ_t^{PT}) is to use the change at time $t-1$ to estimate the future change, i.e., δ_t^{PT}=$h_t - h_{t-1}$. PT is highly straightforward and intuitive. It can work well when h changes approximately linearly at time t.

Sequential Trend (ST) As shown in Figure 2 (b), when the evolution curve is locally similar to a quadratic function, PT has a significant error at the inflection point. This is because it only sees limited historical information, thus cannot accurately estimate the future trend. To alleviate the issue, we develop the sequential trend, which exploits the global information.

Specifically, a state pool (sp) is employed to memorize all historical states, i.e., $sp = \{h_1, h_2, ..., h_{t-1}\}$. Then we denote the change $h_{\tau+1} - h_\tau$ as δ_τ. Finally, the sequential trend δ_t^{ST} is calculated as follows:

$$\delta_t^{ST} = f(\delta_1, \delta_2, ..., \delta_{t-1}). \tag{2}$$

Here, $f(\cdot)$ is the ST function. We hope that δ_t^{ST} equals to the true trend $(h_{t+1} - h_t)$. Therefore, we adopt the following ST loss $l_{st}(t)$:

$$l_{st}(t) = ||\delta_t^{ST} - (h_{t+1} - h_t)||_2^2. \tag{3}$$

[2]Here, Enc and Asm are implemented by convolution layers.

[3]Without ambiguity, the output of Asm is still denoted as h_t.

As for $f(\cdot)$, it can be implemented by a recurrent cell, such as ConvGRU (Ballas et al. 2015). However, ConvGRU tends to retain short-term information in spatiotemporal prediction tasks (Huang et al. 2022a). To better capture the global trend, we develop an attention-based trend unit (ATU), which is shown in Figure 4.

Concretely, an ST state s_{t-1} memorizes the historical trend dynamics, then we calculate the candidate ST state s'_t as follows:

$$\begin{aligned} r_t &= \sigma(W_{r\delta} * \delta_{t-1} + W_{rs} * s_{t-1}) \\ s'_t &= tanh(W_{s\delta} * \delta_{t-1} + r_t \circ (W_{ss} * s_{t-1})) \end{aligned}. \tag{4}$$

Here, r_t is the reset gate. $*$ denotes the convolution operation with W as its parameter.

Then, we use s'_t as the query to extract important information from each historical state:

$$\begin{aligned} Q &= W_q^T * s'_t \\ K_i &= W_{ki}^T * s_i \\ \lambda_i &= softmax(Q^T K_i) \cdot \\ a_t &= \sum_i \lambda_i \cdot s_i \end{aligned} \tag{5}$$

Q and K are the results of query and key, respectively, where W_q and W_k are the parameters. a_t combines all ST states with λ_i as the coefficients. Notice that the early states can play important roles when their coefficients are large. Therefore, ATU can better capture the long-term features.

Finally, we obtain δ_t^{ST} as follows:

$$\begin{aligned} z_t &= \sigma(W_{z\delta} * \delta_{t-1} + W_{zs} * s'_t + W_{za} * a_t) \\ s_t &= (1 - z_t) * s'_t + z_t * a_t \\ \delta_t^{ST} &= conv(s_t) \end{aligned} \quad . \tag{6}$$

Here, z_t is the update gate. $conv(\cdot)$ means the convolutional mapping.

Flow Trend (FT) As for the meteorological elements, information flow is a significant internal cause of their evolution. For example, the heat flows from hot to cold regions,

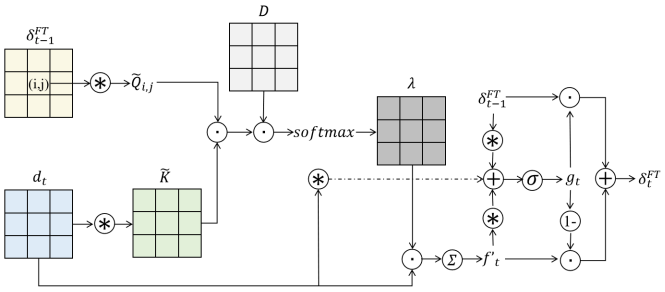


Figure 5: The inner structure of FAU. d_t denotes the local difference. D is a distance-based prior matrix, where the closer regions have large weights.

and the substance flows from high-concentration to low-concentration regions. Inspired by such a common sense, we develop the flow trend (FT).

Specifically, the above flows are usually driven by temperature or concentration difference. Therefore, at a given point $h_{t,i,j}$, we first calculate the difference between it and the surrounding regions, which is shown as follows:

$$d_{t,i',j'} = h_{t,i',j'} - h_{t,i,j}. \tag{7}$$

Here, $(i', j') \in N(i, j)$ is the neighborhood of (i, j).

Then we perform spatial attention on d to capture the flow features, as shown in Figure 5. Here, the previous FT trend δ_{t-1}^{FT} serves as the query. Eq. (8) gives the details:

$$\begin{aligned} \tilde{Q} &= \tilde{W}_q^T * \delta_{t-1}^{FT} \\ \tilde{K}_{i',j'} &= \tilde{W}_k^T * d_{t,i',j'} \\ \tilde{\lambda}_{i',j'} &= softmax\left(D_{i',j'} * \left(\tilde{Q}_{i,j}^T \cdot \tilde{K}_{i',j'}\right)\right) \cdot \\ f_t' &= \sum_{(i',j')} \tilde{\lambda}_{i',j'} \cdot d_{t,i',j'} \end{aligned} \tag{8}$$

$\tilde{Q}$ and $\tilde{K}$ are the results of query and key, with $\tilde{W}_q$ and $\tilde{W}_k$ as the parameters. D is a distance-based prior matrix[4] (Huang et al. 2023).

Finally, the flow trend δ_t^{FT} is calculated as follows:

$$\begin{aligned} g_t &= \sigma(W_{gd} * d_t + W_{gf'} * f_t' + W_{gf} * \delta_{t-1}^{FT}) \\ \delta_t^{FT} &= g_t * \delta_{t-1}^{FT} + (1 - g_t) * f_t' \end{aligned}, \tag{9}$$

where g_t is the update gate with $W_{g\cdot}$ as the parameters.

Trend Fusion Inspired by the selective kernel network (SKNet) (Li et al. 2019), which aims to fuse multiple branches with different kernel sizes, we develop an attention-based fusion module.

Specifically, we first take the average of PT, ST, and FT, where the outcome is denoted as AT. Different from SKNet, we compress AT with a downsampling function instead of a global average pooling. This is because that various trends may play different roles in different positions. Then the outcome CT is utilized to generate the adaptive fusion weight w, which is shown as follows:

$$\begin{aligned} p = us_1(CT), s = us_2(CT), f = us_3(CT) \\ w_p, w_s, w_f = softmax(p, s, f) \end{aligned}. \tag{10}$$

[4]See the Appendix for more details about D.

Here, us is the upsampling function, aiming to map CT back to the original size. w_p, w_s, and w_f are the weights of PT, ST, and FT, respectively.

Finally, we add up the three trends with their weights:

$$\delta_t = w_p * \delta_t^{PT} + w_s * \delta_t^{ST} + w_f * \delta_t^{FT}. \tag{11}$$

Loss Function

The loss function consists of two parts, i.e., the prediction loss l_p and the sequential trend loss l_{st}. (see Eq. (3) for details.) Here, l_p is the commonly used mean square error, which is calculated as follows:

$$l_p(t) = ||u_t - \widehat{u}_t||_2^2. \tag{12}$$

u_t is the true observation and $\widehat{u}_t$ is the prediction. Finally, we obtain the loss l as follows:

$$l = \sum_t (l_p(t) + l_{st}(t)). \tag{13}$$

Interpretable Structures in Our Model

At the end of this section, we summarize the interpretable structures in our model. Specifically, (1) the proposed iTrendRNN exactly follows the differential framework in Eq. (1), which well formulates the meteorological evolution. (2) PT simply uses the most recent changes, so it is exceedingly intuitive. (3) ST is explicitly constrained by l_{st}, aiming to learn the true trends. Furthermore, the temporal attention weights can reveal the roles of different historical states. (4) FT reflects the nature of evolution, namely the flow of matter or energy. The spatial attention weight also indicates the degree of interaction. (5) The trend weights in the fusion module show the importance of each trend.

Experiments

In this paper, we conduct extensive experiments on two important meteorological elements, i.e., atmospheric temperature and relative humidity.

Datasets

The European Centre for Medium-Range Weather Forecasts (ECMWF) provides the global climate and weather data[5]. In our experiment, we focus on the temperature and relative humidity near the ground. Specifically, we select a region in East Asia, with latitude ranging from 20° N to 40° N and longitude ranging from 110° E to 130° E. Here, the spatial resolution is 0.25°. We set the time resolution to 2 hours, and use the previous six frames to predict the next six. The training set consists of 5,840 sequences from 2018 and 2019, and the test set contains 2,912 sequences from 2020.

Implementation Details

The details of the proposed iTrendRNN are listed as follows:

- Input encoder (Enc in Figure 3): It is a two-layer convolutional network, where the channel numbers are 32 and 64, respectively. The kernel sizes are both set to 3.

[5]https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels?tab=overview

- Assimilation module (Asm in Figure 3): It is a one-layer convolution network, where the kernel size is 3.
- Proximity trend module: It is a non-parametric module.
- Sequential trend module: As for Eq. (4), the channel number is 64 and the kernel size is 5. As for Eq. (5), the query and key functions are both implemented by 3×3 convolutions. As for Eq. (6), the kernel size is 5.
- Flow trend module: The query and key functions in Eq. (8) are both implemented by 1×1 convolutions. The neighborhood size is set to 5. Moreover, the kernel size in Eq. (9) is 5.
- Fusion module: The stride and kernel size in downsampling and upsampling functions are 2 and 3, respectively.

Furthermore, our code is publicly available at https://github.com/hub5/iTrendRNN.

Baselines and Evaluation Metrics

We compare our model with the following popular methods: ConvGRU (Ballas et al. 2015), ConvLSTM (Shi et al. 2015), TrajGRU (Shi et al. 2017), PredRNN (Wang et al. 2017), MIM (Wang et al. 2019), PhyDNet (Guen and Thome 2020), Sa-ConvLSTM (Lin et al. 2020), SimVP (Gao et al. 2022b), and TAU (Tan et al. 2023).

Two commonly used metrics are employed to evaluate the prediction performance, i.e., mean square error (MSE) and mean absolute error (MAE). A better prediction is indicated by lower MSE and MAE.

Overall Performance

	Temperature		Relative humidity	
	MSE	MAE	MSE	MAE
ConvGRU	1.32	0.79	43.22	4.54
ConvLSTM	1.25	0.77	43.09	4.50
TrajGRU	1.09	0.70	41.59	4.42
PredRNN	1.15	0.75	40.30	4.35
MIM	1.06	0.70	<u>39.57</u>	<u>4.28</u>
PhyDNet	1.13	0.72	41.15	4.38
SA-ConvLSTM	1.19	0.73	40.51	4.31
SimVP	1.19	0.76	41.91	4.48
TAU	<u>1.04</u>	<u>0.70</u>	40.37	4.35
iTrendRNN	**0.87**	**0.63**	**36.97**	**4.17**

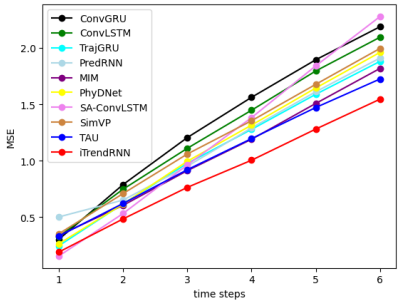
Table 1: Evaluation metrics of all methods.

Table 1 reports the evaluation metrics of all methods. We can see that our iTrendRNN outperforms all the baselines. Specifically, as for the temperature dataset, we improve MSE from 1.04 to 0.87, and MAE from 0.70 to 0.63. As for the relative humidity dataset, we improve MSE from 39.57 to 36.97, and MAE from 4.28 to 4.17.

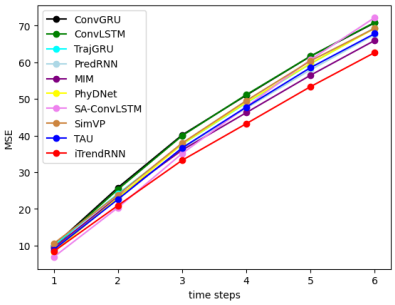
Figure 6 shows the step-wise MSE for two datasets. We can observe that our curves are generally lower than the others. This demonstrates that the proposed iTrendRNN is consistently superior. (See the Appendix for more prediction results.)

Ablation Study

In this part, we evaluate the role of each proposed module.



(a) Temperature.



(b) Relative humidity.

Figure 6: The step-wise MSE for two datasets.

Firstly, we explore the effectiveness of each trend. Specifically, the variants are denoted as $w/o\ PT$, $w/o\ ST$, and $w/o\ FT$, each of which means discarding the corresponding trends. The results are reported in Table 2. We find that: (1) the performance of all variants decreases. This suggests that each trend module helps to make a more accurate prediction. (2) $w/o\ ST$ achieves the worst metrics, which indicates that ST is the most important.

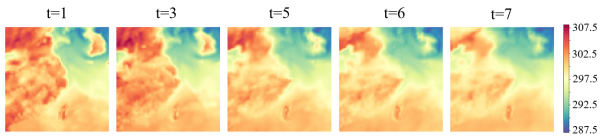
	Temperature		Relative humidity	
	MSE	MAE	MSE	MAE
$w/o\ PT$	0.98	0.69	39.09	4.32
$w/o\ ST$	1.46	0.82	45.82	4.73
$w/o\ FT$	1.06	0.72	40.17	4.37
iTrendRNN	**0.87**	**0.63**	**36.97**	**4.17**

Table 2: Evaluation metrics of all trend variants.

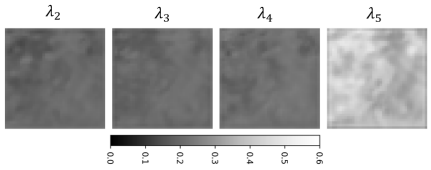
Secondly, we explore the effectiveness of trend fusion module. Here, we compare it with the other three fusion ways, i.e., average fusion (AF), convolution fusion CF, and SKNet fusion (SF). The results are reported in Table 3. We can see that our fusion strategy yields the best results. Particularly, compared with SF, our method is better because it takes into account that various trends may play different roles in different regions.

	Temperature		Relative humidity	
	MSE	MAE	MSE	MAE
AF	0.94	0.66	39.15	4.32
CF	0.92	0.65	38.26	4.28
SF	0.89	0.64	37.89	4.24
iTrendRNN	**0.87**	**0.63**	**36.97**	**4.17**

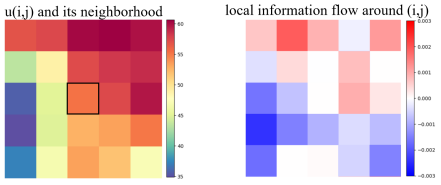
Table 3: Evaluation metrics of all fusion variants.



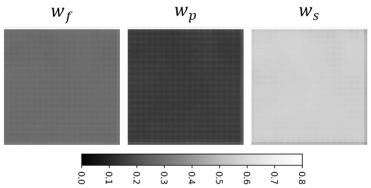
(a) The data sequence. Here, we are predicting the 7-th frame.



(b) Temporal attention weights. According to Figure 4, we are esimating δ_6^{ST}. Hence, the historical states include s_2, s_3, s_4, and s_5.

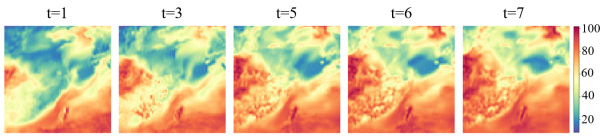


(c) Information flow. According to Eq. (8), we show the flow with $(\tilde{\lambda}_{i',j'} \cdot d_{t,i',j'})$.

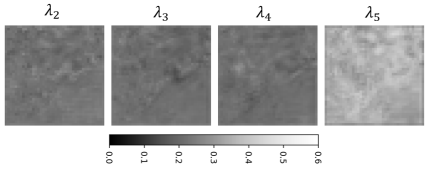


(d) Trend attention weights. w_f, w_p, and w_s are for the flow trend, proximity trend, and sequential trend, respectively.

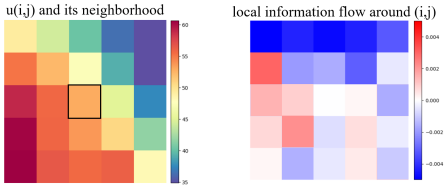
Figure 7: Attention analysis on a temperature sample.



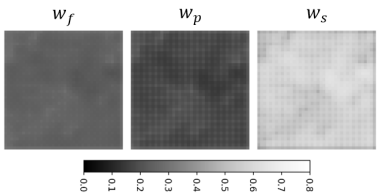
(a) The data sequence. Here, we are predicting the 7-th frame.



(b) Temporal attention weights. According to Figure 4, we are esimating δ_6^{ST}. Hence, the historical states include s_2, s_3, s_4, and s_5.



(c) Information flow. According to Eq. (8), we show the flow with $(\tilde{\lambda}_{i',j'} \cdot d_{t,i',j'})$.



(d) Trend attention weights. w_f, w_p, and w_s are for the flow trend, proximity trend, and sequential trend, respectively.

Figure 8: Attention analysis on a relative humidity sample.

Attention Analysis

In this part, we analyze the attention mechanisms used in iTrendRNN.

Firstly, we visualize the temporal attention in ST. Figures 7(b) and 8(b) show the learned weights. We can see that the most recent state s_5 has the largest weight λ_5. On the other hand, λ_2, λ_3, and λ_4 also play a nonnegligible role, which indicates the importance of long-term features.

Secondly, we visualize the spatial flow attention in FT. Figures 7(c) and 8(c) show the learned information flow. Specifically, the left subfigure depicts the values of a given position (i, j) and its neighborhood. The right subfigure shows the information flow from its neighborhood. We can find that: (1) as for Figure 7(c), the temperature of the upper part is higher than that of (i, j), and the corresponding flow is also generally positive. The phenomenon in the left part is exactly the opposite. (2) As for Figure 8(c), something similar happens. For example, the humidity of the upper part is lower than that of (i, j), and the corresponding flow is also generally negative.

Lastly, we visualize the attention in trend fusion module. Figures 7(d) and 8(d) show the learned weights. We can observe that: (1) ST has the largest weights, which means it is the most important part. (2) In Figure 8(d), the weight distribution in the spatial domain shows certain differences. This suggests that various trends play different roles in different positions, thus we improve the fusion way in SKNet. (See the Appendix for more attention analysis.)

Conclusion

In this paper, we propose an interpretable trend-aware RNN, named iTrendRNN, for meteorological spatiotemporal prediction tasks. iTrendRNN explicitly adopts a differential prediction framework, where the key lies in estimating the evolutionary trends. Herein, three types of trends are exploited, i.e., proximity trend (PT), sequential trend (ST), and flow trend (FT). All of them can be well explained. Furthermore, we develop an attention-based trend unit for ST, aiming to better capture long-term features. We also design a flow-aware attention unit for FT, which can reflect the local information interactions. Finally, a trend fusion module is employed to adaptively fuse the three trends. Extensive experiments on two datasets show that our iTrendRNN outperforms existing methods.

Acknowledgments

This work was supported in part by National Nature Science Foundation of China (No. 62272130, 62376072, and 62306184), Shenzhen Science and Technology Program (No. KCXFZ20211020163403005), Nature Science Program of Shenzhen (No. JCYJ20210324120208022), and Natural Science Foundation of Top Talent of SZTU (grant no. GDRC202320).

References

Arras, L.; Arjona-Medina, J.; Widrich, M.; Montavon, G.; Gillhofer, M.; Müller, K.-R.; Hochreiter, S.; and Samek, W. 2019. Explaining and interpreting LSTMs. *Explainable ai: Interpreting, explaining and visualizing deep learning*, 211–238.

Ballas, N.; Yao, L.; Pal, C.; and Courville, A. 2015. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*.

Barić, D.; Fumić, P.; Horvatić, D.; and Lipic, T. 2021. Benchmarking attention-based interpretability of deep learning in multivariate time series predictions. *Entropy*, 23(2): 143.

Belharbi, S.; Sarraf, A.; Pedersoli, M.; Ben Ayed, I.; McCaffrey, L.; and Granger, E. 2022. F-cam: Full resolution class activation maps via guided parametric upscaling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3490–3499.

Chien, J.-T.; and Chen, Y.-H. 2021. Continuous-time attention for sequential learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 7116–7124.

Dai, F.; Huang, P.; Mo, Q.; Xu, X.; Bilal, M.; and Song, H. 2022a. ST-InNet: Deep Spatio-Temporal Inception Networks for Traffic Flow Prediction in Smart Cities. *IEEE Transactions on Intelligent Transportation Systems*, 23(10): 19782–19794.

Dai, K.; Li, X.; Ye, Y.; Feng, S.; Qin, D.; and Ye, R. 2022b. MSTCGAN: Multiscale time conditional generative adversarial network for long-term satellite image sequence prediction. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–16.

Gao, Z.; Shi, X.; Wang, H.; Zhu, Y.; Wang, Y. B.; Li, M.; and Yeung, D.-Y. 2022a. Earthformer: Exploring space-time transformers for earth system forecasting. *Advances in Neural Information Processing Systems*, 35: 25390–25403.

Gao, Z.; Tan, C.; Wu, L.; and Li, S. Z. 2022b. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3170–3180.

Guen, V. L.; and Thome, N. 2020. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11474–11484.

He, Z.; Chow, C.-Y.; and Zhang, J.-D. 2020. STNN: A spatio-temporal neural network for traffic predictions. *IEEE Transactions on Intelligent Transportation Systems*, 22(12): 7642–7651.

Hou, B.-J.; and Zhou, Z.-H. 2020. Learning with interpretable structure from gated RNN. *IEEE transactions on neural networks and learning systems*, 31(7): 2267–2279.

Huang, X.; Li, X.; Ye, Y.; Feng, S.; Luo, C.; and Zhang, B. 2022a. On understanding of spatiotemporal prediction model. *IEEE Transactions on Circuits and Systems for Video Technology*.

Huang, X.; Zhang, B.; Feng, S.; Ye, Y.; and Li, X. 2023. Interpretable local flow attention for multi-step traffic flow prediction. *Neural networks*, 161: 25–38.

Huang, X.; Zhang, B.; Ye, Y.; Feng, S.; and Li, X. 2022b. Spatiotemporal prediction in three-dimensional space by separating information interactions. *Applied Intelligence*, 1–14.

Kuligowski, R. J.; and Barros, A. P. 1998. Localized precipitation forecasts from a numerical weather prediction model using artificial neural networks. *Weather and forecasting*, 13(4): 1194–1204.

Li, X.; Wang, W.; Hu, X.; and Yang, J. 2019. Selective kernel networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 510–519.

Lin, Z.; Li, M.; Zheng, Z.; Cheng, Y.; and Yuan, C. 2020. Self-attention convlstm for spatiotemporal prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 11531–11538.

Ma, F.; Hu, X.; Liu, A.; Yang, Y.; Philip, S. Y.; Wen, L.; et al. 2023. AMR-based Network for Aspect-based Sentiment Analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 322–337.

Munkhdalai, L.; Munkhdalai, T.; Park, K. H.; Amarbayasgalan, T.; Batbaatar, E.; Park, H. W.; and Ryu, K. H. 2019. An end-to-end adaptive input selection with dynamic weights for forecasting multivariate time series. *IEEE Access*, 7: 99099–99114.

Nauta, M.; Schlötterer, J.; van Keulen, M.; and Seifert, C. 2023. PIP-Net: Patch-Based Intuitive Prototypes for Interpretable Image Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2744–2753.

Papi, S.; Negri, M.; and Turchi, M. 2022. Attention as a guide for Simultaneous Speech Translation. *arXiv preprint arXiv:2212.07850*.

Peng, J.; Xiong, Z.; Tan, H.; Huang, X.; Li, Z.-P.; and Xu, F. 2022. Boosting photon-efficient image reconstruction with a unified deep neural network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 4180–4197.

Peng, Z.; and Huang, X. 2022. Spatial-temporal transformer network with self-supervised learning for traffic flow prediction.

Ravuri, S.; Lenc, K.; Willson, M.; Kangin, D.; Lam, R.; Mirowski, P.; Fitzsimons, M.; Athanassiadou, M.; Kashem, S.; Madge, S.; et al. 2021. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878): 672–677.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28.

Shi, X.; Gao, Z.; Lausen, L.; Wang, H.; Yeung, D.-Y.; Wong, W.-k.; and Woo, W.-c. 2017. Deep learning for precipitation nowcasting: A benchmark and a new model. *Advances in neural information processing systems*, 30.

Tan, C.; Gao, Z.; Wu, L.; Xu, Y.; Xia, J.; Li, S.; and Li, S. Z. 2023. Temporal attention unit: Towards efficient spatiotemporal predictive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18770–18782.

Tran, D. T.; Iosifidis, A.; Kanniainen, J.; and Gabbouj, M. 2018. Temporal attention-augmented bilinear network for financial time-series data analysis. *IEEE transactions on neural networks and learning systems*, 30(5): 1407–1418.

Wang, Y.; Long, M.; Wang, J.; Gao, Z.; and Yu, P. S. 2017. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems*, 30.

Wang, Y.; Zhang, J.; Zhu, H.; Long, M.; Wang, J.; and Yu, P. S. 2019. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9154–9162.

Wu, H.; Yao, Z.; Wang, J.; and Long, M. 2021. MotionRNN: A flexible model for video prediction with spacetime-varying motions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15435–15444.

Xia, M.; Jin, D.; and Chen, J. 2022. Short-term traffic flow prediction based on graph convolutional networks and federated learning. *IEEE Transactions on Intelligent Transportation Systems*, 24(1): 1191–1203.

Xu, Z.; Wang, Y.; Long, M.; Wang, J.; and KLiss, M. 2018. PredCNN: Predictive Learning with Cascade Convolutions. In *IJCAI*, 2940–2947.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, 818–833. Springer.

Zhang, J.; Zheng, Y.; and Qi, D. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Zhang, Q.; Cheng, X.; Chen, Y.; and Rao, Z. 2022. Quantifying the knowledge in a DNN to explain knowledge distillation for classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 5099–5113.

Zhang, Q.; Yang, Y.; Ma, H.; and Wu, Y. N. 2019. Interpreting cnns via decision trees. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6261–6270.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.