

# Imputing Missing Temperature Data of Meteorological Stations Based on Global Spatiotemporal Attention Neural Network

Tianrui Hou<sup>1</sup>[0009-0000-6474-957X], Xinshuai Guo<sup>1</sup>, Li Wu<sup>1</sup>[0000-0001-5242-3169], Xiaoying Wang<sup>1</sup>[0000-0003-1029-0358], Guojing Zhang<sup>1</sup>, and Jianqiang Huang<sup>1</sup>

<sup>1</sup>Qinghai University, Qinghai QHU 10743, CHN

**Abstract.** Imputing missing meteorological site temperature data is necessary and valuable for researchers to analyze climate change and predict related natural disasters. Prior research often used interpolation-based methods, which basically ignored the temporal correlation existing in the site itself. Recently, researchers have attempted to leverage deep learning techniques. However, these models cannot fully utilize the spatiotemporal correlation in meteorological stations data. Therefore, this paper proposes a global spatiotemporal attention neural network (GSTA-Net), which consists of two sub networks, including the global spatial attention network and the global temporal attention network, respectively. The global spatial attention network primarily addresses the global spatial correlations among meteorological stations. The global temporal attention network predominantly captures the global temporal correlations inherent in meteorological stations. To further fully exploit and utilize spatiotemporal information from meteorological station data, adaptive weighting is applied to the outputs of the two sub-networks, thereby enhancing the imputation performance. Additionally, a progressive gated loss function has been designed to guide and accelerate GSTA-Net's convergence. Finally, GSTA-Net has been validated through a large number of experiments on public dataset TND and QND with missing rates of 25%, 50%, and 75%, respectively. The experimental results indicate that GSTA-Net outperforms the latest models, including Linear, NLinear, DLinear, PatchTST, and STA-Net, across both the mean absolute error (MAE) and the root mean square error (RMSE) metrics.

**Keywords:** Attention mechanism, Deep learning, Neural network, Missing data imputing, Meteorological station data, Spatiotemporal correlation.

## 1. Introduction

In the field of meteorology, most accurate temperature data comes from meteorological observation stations and is collected by temperature sensors. Complete temperature data can improve the accuracy of meteorologists in weather forecasting and climate analysis, and is also an important source of data in agricultural and ecological disaster research [1-2,4]. However, due to various issues such as electromagnetic

interference, equipment failures, harsh environmental conditions, or manual operation errors, the data collected by meteorological stations is not always complete [1,3-4], which hinders relevant scientific research. Therefore, reconstructing or imputing missing temperature data is an essential preliminary task in conducting related scientific work and is a problem that urgently needs to be addressed.

Early models used to fill in missing temperature data mostly were based on interpolation methods, which could not fully utilize the temporal correlation of meteorological station data itself. With the development of machine learning, researchers have used models such as expectation maximization (EM) [5], multiple regression (MR) [6], and Bayesian networks [7] to fill in missing values. However, these models can uncover the complex relationships or potential distributions present in meteorological station data, leading to hardly utilizing spatiotemporal information within meteorological stations data. Although deep learning is a branch of machine learning, deep learning has the ability to extract complex features and correlations from large amounts of data [8]. Meanwhile, due to meteorological station data are time-series data, researchers have used frameworks based on Long Short-Term Memory (LSTM) networks [9] to capture the potential temporal features of station data to fill in missing data. However, LSTM based models only handle the temporal correlation of individual meteorological stations and cannot effectively utilize the spatial correlation between meteorological stations, leading to insufficient imputing performance. Recently, STA-Net proposed by Hou et al. [10] successfully used CNN to capture the spatial information between stations, improving the imputing performance. But this model still lacked extraction of global spatiotemporal information in data.

Faced with the above-mentioned dilemma, this paper designed a novel global spatiotemporal attention neural network (GSTA-Net), based on the research of Hou et al. [10]. The core components of GSTA-Net include feature expression model (FEM) used to generate high-dimensional feature vectors, global temporal self-attention mechanism used to obtain global temporal information, and global spatial self-attention mechanism used to obtain global spatial information. Additionally, to further accelerate the convergence speed of GSTA-Net and reconstruction performance, we also designed a new loss function—progressive gated loss function. Meanwhile, we conducted abundant experiments on real meteorological datasets—TND and QND [10]. Results demonstrated that designed GSTA-Net outperformed the latest models. In summary, the main contributions of this article are as follows:

- developing the GSTA-Net, which is a model designed to reconstruct missing data from meteorological stations and can effectively harnesses global spatiotemporal information, as demonstrated by its enhanced imputing accuracy in experimental evaluations;
- introducing a Feature Expression Model (FEM) that transforms low-dimensional meteorological data into high-dimensional feature vectors, and on its basis integrating the global temporal and spatial self-attention modules, which work in tandem to effectively capture and process global spatiotemporal relationships within the meteorological station data;

- innovating a novel loss function known as the progressive gated loss function, which is designed to steer the model's convergence and refine the learning process of the FEM, allowing it to effectively distinguish and exclude noise that is irrelevant to the global spatiotemporal context.

## 2. Related work

Previous research on modeling missing values in meteorological station data can generally be divided into two categories: spatial interpolation methods and data mining-based machine learning methods.

Space based interpolation methods are according to the mathematical or physical properties contained in meteorological station data, and fill in missing values through mathematical statistical analysis [11]. They typically include inverse distance weighting (IDW) [12], kriging [13], and thin plate splines [14] and etc. However, because of sensitivity to data fluctuations, spatial distance and the number of meteorological stations [3], they show a significant performance decline on higher missing rate. Even though these methods perform well and fill quickly on low missing rate.

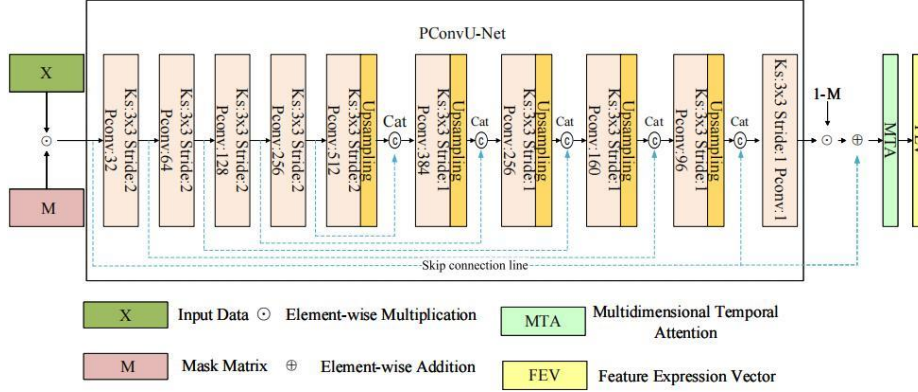
To solve above-mentioned problem, researchers started using machine learning to mine complex relationships in meteorological station data to fill in missing values. Generally, these methods embody EM [5], (MR) [6], Bayesian networks [7], artificial neural network (ANN) [15] and so on. Although such methods have better missing value reconstruction performance compared to interpolation methods, they cannot fully utilize the spatiotemporal correlations in meteorological station data and are highly sensitive to data [16], causing a serious decrease in filling accuracy on high missing rate.

To solve above-mentioned problem, researchers have started to use deep learning to deal with this issue. For example, Xie et al. [17] used Bi-LSTM to fill in missing station temperature data, effectively utilizing the correlations in time series data. But it was difficult to capture long-distance dependencies. Nie et al. [18] proposed PatchTST to carry out long-term prediction. Although PatchTST solved the global dependencies by making use of Transformer [19], it lacked the utilization of spatial information. The linear model proposed by Zeng et al. [20] was a time series data prediction model, and was not a Transformer architecture. Even so, this model had high computational efficiency and could ensure the prediction accuracy of long sequence data, it was not suitable for a shorter sequence data. However, the aforementioned models rarely utilized spatiotemporal information in meteorological station data.

Recently, the STA-Net proposed by Hou et al. [10] addressed this issue. Nevertheless, this model only considers local spatiotemporal correlations, neglecting global spatiotemporal information. To solve faced problem, this paper designs a global spatiotemporal attention neural network (GSTA-Net).

### 3. Methods

#### 3.1 Feature Expression Model



**Fig. 1.** Illustration of feature expression model. It can convert low dimensional masked data into high-dimensional feature vectors.

In the field of deep learning, in order to make the relevant research tasks proceed smoothly, it is first necessary to obtain the abstract feature vector representation of the research object [21]. Presently, in computer vision (CV) and natural language processing (NLP), mostly using the encoder in the encoder-decoder model structure to obtain advanced abstract feature representations of data, and then performing subsequent downstream tasks [22-23]. Inspired by this, after in-depth analysis of the latest model STA-Net [10], a Feature Expression Model (FEM) was extracted from it, as shown in Fig. 1. The main function of FEM is to abstract the temperature data of meteorological stations at different times into high-dimensional feature vector representations.

FEM consists of partial convolutional neural network (PConvU-Net) [24] and multidimensional temporal attention (MTA) [10]. Among them, the input data of PConvU-Net is the hadamard product  $X_m \in \mathbb{R}^{1 \times h \times w}$  of the real data  $X \in \mathbb{R}^{1 \times h \times w}$  and the corresponding mask  $M \in \mathbb{R}^{1 \times h \times w}$ , where  $h$  and  $w$  represent the row and column dimensions of  $X$ . PConvU-Net is composed of 10 layers of PConv [24] whose convolution core ( $Ks$  in Fig. 1.) is set to  $3 \times 3$ , where the first 5 layers form the encoder layer and the last 5 layers form the decoder layer. In encoder, set the sliding step (*Stride* in Fig. 1.) to 2; in decoder layer, the sliding step is set to 1. In addition, an up-sampling layer is used in front of each layer in decoder to increase the dimensionality of the data, facilitating concatenation with the output of the corresponding layer of the Encoder. At the same time, the skip connection line used by PConvU-Net to concatenate the corresponding feature maps of the Encoder and Decoder layers, not only do facilitate network learning and convergence, but also alleviates vanishing or exploding gradients. The function of this module is to preliminarily explore potential spatial information in the data and fill in missing data in space. The input

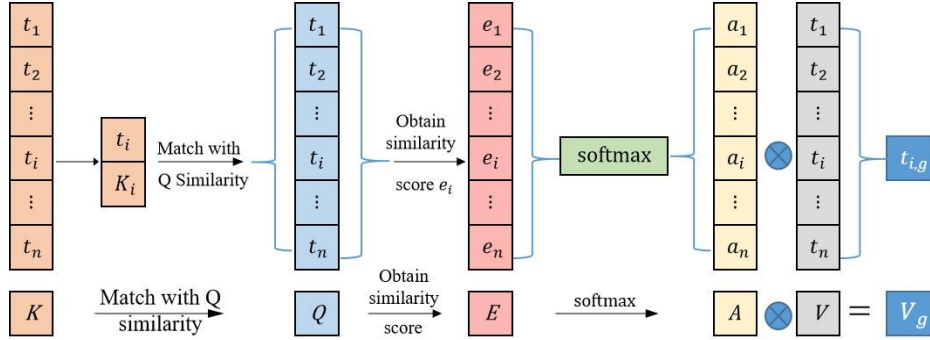
data of the MTA is the feature vector  $\tilde{X} \in \mathbb{R}^{h \times w \times 1}$ . To obtain  $\tilde{X}$ , need to reshape the result fusing  $X$  and the output of PConvU-Net. The main function of this module include: 1) obtaining preliminary temporal information of data; 2) feature vector dimension transformation, that is, transforming low dimensional feature vectors into high-dimensional feature vectors. In summary, the function of FEM can be defined by the following equation (1) and (2):

$$\tilde{X} = R(G_p(X \odot M; \theta_p) \odot (1 - M) + X \odot M) \quad (1)$$

$$FEV = G_{MAT}(\tilde{X} \odot M; \theta_{MAT}) \quad (2)$$

wherein  $R$  denotes the remodeling function,  $G_p$  and  $G_{MAT}$  are PConvU-Net and MAT, respectively.  $\theta_p$  and  $\theta_{MAT}$  are their learnable parameters.  $\odot$  denotes element-wise multiplication.  $FEV$  denotes needed feature vector representations. In summary, FEM transforms low dimensional input data features  $X$  into high-dimensional feature expression vectors  $FEV \in \mathbb{R}^{c \times h \times w}$ , which not only increases information volume but also preliminarily integrates spatiotemporal information, being beneficial for obtaining complex spatiotemporal information from data in subsequent tasks. Wherein  $c$  represents the dimension of  $FEV$

### 3.2 Global Temporal and Spatial Self-Attention Mechanism



**Fig. 2.** Illustration of global temporal self-attention mechanism. Through this mechanism, making the feature expression vector at any time in sequence contain global temporal information.

**Global Temporal Self-Attention Mechanism.** The feature expression vector acquired by FEM can indicate temperature data of meteorological stations at different times. Intuitively, the temperature trends of any meteorological station at adjacent times are similar. Besides, temperature data is also a highly periodic temporal data. That is, there is a certain degree of similarity between the feature expression vectors of adjacent moments close periods. Accordingly, we design the global temporal self-attention mechanism (GTSA), as shown in Fig. 2.

In the GTSA, its input sequence data is named and defined as  $S = (\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_i, \dots, \mathbf{t}_{n-1}, \mathbf{t}_n)$ , wherein  $n$  is the length of  $S$  and  $\mathbf{t}_i$  is the high-dimensional vector. To expediently study the global relationships between  $\mathbf{t}_i$  and  $S$ , abstract  $S$  into three tensors: Key  $\mathbf{K}$ , Query  $\mathbf{Q}$  and Value  $\mathbf{V}$ . As shown in Fig. 2, first select feature expression vector  $K_i$  from  $\mathbf{K}$  to perform similarity-matching operation with all vectors in  $\mathbf{Q}$ , which will attain the similarity score  $E$ . This process is defined as equation (3):

$$E = v^T(L_k(\mathbf{K}; \theta_k) + L_q(\mathbf{Q}; \theta_q)) \quad (3)$$

wherein  $v$  is learnable feature vector,  $L_k$  and  $L_q$  are two linear neural networks, and  $\theta_k$  and  $\theta_q$  are the learnable parameters of  $L_k$  and  $L_q$ , respectively.  $E$  indicates how much the feature vector at different moments is similar to other all feature expression vector. Then, to acquiring  $V_g$  containing global temporal information, carry out matrix multiplication on attention score  $\mathbf{A}$  and  $\mathbf{V}$ , where  $\mathbf{A}$  is come by executing *softmax* on  $E$ . This process is defined as equation (4) and (5):

$$\mathbf{A} = \text{softmax}(E) \quad (4)$$

$$V_g = \mathbf{A}\mathbf{V} \quad (5)$$

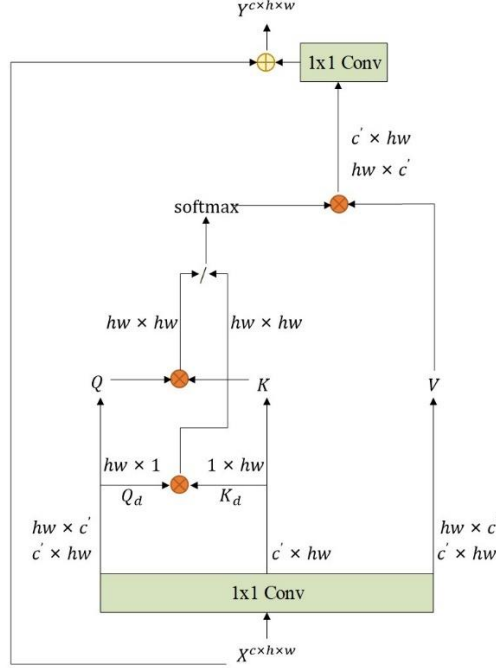
**Global Spatial Self-Attention Mechanism.** For a certain meteorological station and other stations close to it, they have similar temperature and variation patterns, within adjacent time and space [25]. Furthermore, since the  $X$  is a two-dimensional numerical matrix, it can be regarded as single-channel image data. And previous studies have shown that a pixel in an image can maintain continuity and consistency with other pixels in its neighboring space in most cases [26]. Meanwhile, after extensive investigation and research, it was found that the DNL-Net [27] was consistent with the aforementioned rules. Thus, this paper proposed and designed a global spatial self-attention mechanism (GSSA) based on DNL-Net [27], as shown in Fig. 3.

The GSSA works as follows: let the input data of the global spatial self-attention mechanism be  $FEV \in \mathbb{R}^{c \times h \times w}$ .  $FEV$  first goes through a layer CNN whose convolution kernel is  $1 \times 1$ , not undergoes three different CNN layers like in the original DNL-Net. This is because our research has found that using a layer of CNN can ensure information consistency and reduce computational operations. Through this operation, we obtained the output  $FEV' \in \mathbb{R}^{c' \times h \times w}$ . That is, by setting the dimension  $c'$  to avoid the information redundancy or insufficiency due to  $FEV$  is too large or too small. Here, Then, in order to obtain the correlation between a certain meteorological station and other meteorological stations in the nearby space and adjacent time periods, we selected the self-attention mechanism proposed by [19]. To obtain  $\mathbf{K}$ ,  $\mathbf{Q}$ , and  $\mathbf{V}$ , we carried out equation (6) and (7) on  $FEV'$ , as shown as follows:

$$\mathbf{K} = R(FEV') \quad (6)$$

$$\mathbf{Q} = \mathbf{K} = T(R(FEV')) \quad (7)$$

wherein  $R$  represents reshape function and  $T$  represents transpose function. Just, the  $FEV'$  was turned into  $\mathbf{K} \in \mathbb{R}^{c' \times hw}$ ,  $\mathbf{Q} \in \mathbb{R}^{hw \times c'}$  and  $\mathbf{V} \in \mathbb{R}^{hw \times c'}$ . Next, use cosine



**Fig. 3.** Illustration of global spatial self-attention mechanism. Through this mechanism, making the feature expression vector contain global spatial information.

similarity instead of dot product to perform global spatial correlation matching on  $\mathbf{K}$  and  $\mathbf{Q}$ . Not selecting dot product in the original DNL-Net is due to it essentially belongs to vector product, which cannot accurately indicate the angle between two vectors, leading to not accurately reflecting the trend of data change. Performing the cosine similarity operation, we need to calculate the vector length  $\mathbf{K}_d \in \mathbb{R}^{1 \times hw}$  and  $\mathbf{Q}_d \in \mathbb{R}^{hw \times 1}$  of  $\mathbf{K}$  and  $\mathbf{Q}$  in the  $c'$  dimension, as shown in equation (8) and (9):

$$\mathbf{K}_d = \sqrt{\sum_{i=1}^{c'} c_{i,k}'^2} \quad (8)$$

$$\mathbf{Q}_d = \sqrt{\sum_{i=1}^{c'} c_{i,q}'^2} \quad (9)$$

After obtaining  $\mathbf{K}_d$  and  $\mathbf{Q}_d$ , we can calculate the **Attention** of  $\mathbf{K}$ ,  $\mathbf{Q}$  and  $\mathbf{V}$ , as shown in equation (10):

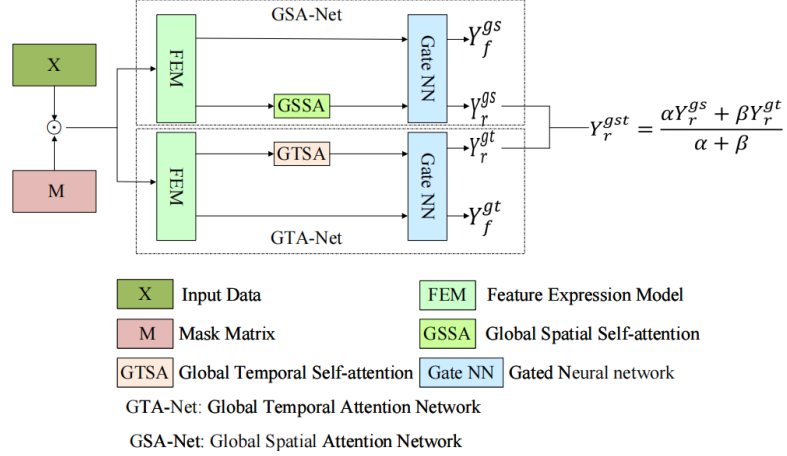
$$\mathbf{Attention}(\mathbf{K}, \mathbf{Q}, \mathbf{V}) = softmax\left(\frac{\mathbf{QK}}{\mathbf{Q}_d \mathbf{K}_d}\right) \mathbf{V} \quad (10)$$

Obviously,  $\mathbf{Attention}(\mathbf{K}, \mathbf{Q}, \mathbf{V})$  contains the global spatial correlation between weather stations. Finally, in order to prevent performance degradation of the original model after undergoing GSSA, residual operation [28] was used. Specifically,  $\mathbf{Attention}(\mathbf{K}, \mathbf{Q}, \mathbf{V})$  successively goes through the transpose function  $T$  and the reshape function  $R$ , and then is input to a convolutional neural network  $F_{conv}$  with a convolution kernel of  $1 \times 1$ , obtaining the output in the same dimension as the input feature data  $X$ . Add the output and  $X$  to acquire final result  $Y$ , as shown in equation (11):

$$Y = X + F_{conv}(R(T(\mathbf{Attention}(\mathbf{K}, \mathbf{Q}, \mathbf{V}))); \theta_{conv}) \quad (11)$$

wherein  $\theta_{conv}$  is learnable parameter.

### 3.3 Global Spatiotemporal Attention Model



**Fig. 4.** Illustration of global spatiotemporal attention neural network. Through it, spatiotemporal information in the data can be fully utilized.

On the basis of the first two sections, we design the global spatiotemporal attention neural network (GSTA-Net), which is composed of two sub networks: global temporal attention network (GTA-Net) and global spatial attention network (GSA-Net). Each sub network consists of FEM, corresponding global attention, and gated neural networks. The architecture of GSTA-Net is shown in the Fig. 4. The following paragraph will describe its execution process.

It can be found from Fig. 4 that the input data of GSTA-Net is the same as FEM, namely is  $X_m$ . When training GSTA-Net,  $X_m$  first passes through GSA-Net and GTA-Net in sequence to obtain the corresponding outputs. And then to obtain the final output result, fusing the output results of the two sub networks. For GSA-Net or GTA-Net,  $X_m$  first goes through FEM to get the feature expression vector ( $FEV$ ). Then  $FEV$  passes through corresponding global attention and gated neural networks in parallel, outputting two feature map with and without global information. In paper,



using linear neural networks to represent gated neural networks, whose function is to regulate the learning process of FEM and filter out components or noise in  $FEV$  that are not related to global temporal or spatial information. In short, when  $X_m$  passes through GTA-Net, outputting  $Y_r^{gt}$  with global temporal information and  $Y_f^{gt}$  without global temporal information. Similarly, when  $X_m$  passes through GSA-Net, outputting  $Y_r^{gs}$  with global temporal information and  $Y_f^{gs}$  without global temporal information. This process can be defined as equations (12), (13), (14) and (15):

$$Y_r^{gt} = L_t(G_t(FEM_t(X \odot M; \theta_{ft}); \theta_{gt}); \theta_{lt}) \quad (12)$$

$$Y_{rf}^{gt} = L_t(FEM_t(X \odot M; \theta_{ft}); \theta_{lt}) \quad (13)$$

$$Y_r^{gs} = L_s(G_s(FEM_s(X \odot M; \theta_{fs}); \theta_{gs}); \theta_{ls}) \quad (14)$$

$$Y_{rf}^{gs} = L_s(FEM_s(X \odot M; \theta_{fs}); \theta_{ls}) \quad (15)$$

wherein  $L_t$  and  $L_s$  are gated neural networks,  $FEM_t$  and  $FEM_s$  are FEM of GTA-Net and GSA-Net, respectively.  $G_t$  and  $G_s$  denote GTA-Net and GSA-Net, separately.  $\theta_{lt}$  and  $\theta_{ls}$  are corresponding learnable parameter.  $\theta_{lt}$ ,  $\theta_{gt}$ ,  $\theta_{ft}$ ,  $\theta_{ls}$ ,  $\theta_{gs}$  and  $\theta_{fs}$  are learnable parameters of the corresponding network.

To fuse the outputs of two sub networks and obtain outputs  $Y_r^{gst}$  with global spatio-temporal information, we used the adaptive weighting formula, as shown as equations (16):

$$Y_r^{gst} = \frac{\alpha Y_r^{gt} + \beta Y_r^{gs}}{\alpha + \beta} \quad (16)$$

wherein  $\alpha$  and  $\beta$  are learnable parameters, used for adjusting  $Y_r^{gt}$  and  $\beta Y_r^{gs}$ .

### 3.4 Progressive Gated Loss Function

Being aimed at designed GSTA-Net, we propose a progressive gated loss function  $Loss_{pg}$ , whose core principle stems from gated convolution recurrent neural networks [29]. It mainly is used for accelerating GSTA-Net convergence and guiding the learning process of FEM. To get  $Loss_{pg}$ , we design following equations:

$$Loss_f^{gt} = \|(X - Y_f^{gt}) \odot (1 - M)\|^2 \quad (17)$$

$$Loss_r^{gt} = \|(X - Y_r^{gt}) \odot (1 - M)\|^2 \quad (18)$$

$$Loss_f^{gs} = \|(X - Y_f^{gs}) \odot (1 - M)\|^2 \quad (19)$$

$$Loss_r^{gs} = \|(X - Y_r^{gs}) \odot (1 - M)\|^2 \quad (20)$$

$$Loss_{pg} = \gamma \left(1 - \frac{e}{e_{max}}\right) (Loss_f^{gt} + Loss_f^{gs}) + Loss_r^{gt} + Loss_r^{gs} \quad (21)$$

wherein  $Loss_f^{gt}$  and  $Loss_f^{gs}$  represent the loss values without global temporal information and global spatial information, respectively.  $Loss_r^{gt}$  and  $Loss_r^{gs}$  represent the loss values with global temporal information and global spatial information, respectively.  $e$  and  $e_{max}$  represent iterations and maximum iterations in learning process.  $\gamma$  It is a hyperparameter, used to control the contributions of  $Loss_f^{gt}$  and  $Loss_f^{gs}$  and set to 4. According to the equation (21), it can be inferred that as iterations increases, the contributions of  $Loss_f^{gt}$  and  $Loss_f^{gs}$  to the entire loss  $Loss_{pg}$  gradually decrease. This means that the gated neural network gradually adjusts the learning process of the FEM, making  $FEV$  gradually reduce components or noise unrelated to global temporal or spatial information.

## 4. Experiments

### 4.1 Experimental Setup and Evaluation Indicators

To verify GSTA-Net, we conducted extensive experiments on two real datasets: TND and QND [10], both of which are single channel numerical matrix data. TND contains 2918 data samples and dimensions of each sample are (24, 61). QND contains 4980 data samples and dimensions of each sample are (24, 37). During the experiment, they were divided into two parts: the training set (accounting for 80% of the total) and the testing set (accounting for 20% of the total). The mask dataset used in the experiment came from literature [11].

All experiments were conducted on a server configured with 8 Nvidia P100 GPUs, Intel (R) Xeon (R) Silver 4216 CPUs @ 2.10GHz, and 256GB of memory. GSTA-Net was implemented on the deep learning framework Pytorch 1.8.1. During the experiment, the epoch, the initial learning rate and batch size were set to 600, 0.001, and 32 separately. Hyperparameter  $c'$  was set to 512. To optimize the training of the entire model, we first trained two sub models separately and saved their parameters. Then, we loaded the trained parameters into the entire model and performed fine-tuning, where initial learning rate was set to  $10^{-6}$ .

In the experiment, mean absolute error (MAE) and root mean square error (RMSE) were used as validation indicators. The relevant calculation formulas are as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - y_i| \quad (22)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2} \quad (23)$$

wherein  $x_i$  and  $y_i$  represent the true and predicted values, respectively.  $\bar{x}$  is average and  $n$  is sample size.

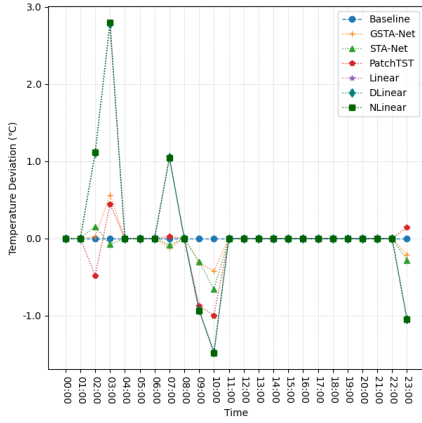
## 4.2 Analysis of Experimental Results

**Table 1.** Results of different models on the TND

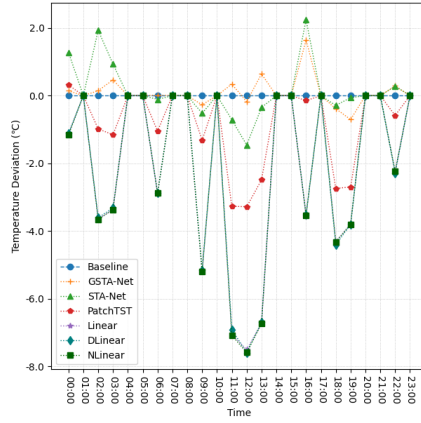
Missing Rate	Methods	MAE ( $\downarrow$ )	RMSE ( $\downarrow$ )
25%	Linear	0.3383	0.8920
	DLinear	0.3421	0.9031
	NLinear	0.3387	0.8892
	PatchTST	0.1615	0.4575
	STA-Net	0.1430	0.4093
	GSTA-Net	<b>0.1361</b>	<b>0.3913</b>
50%	Linear	0.8116	1.4994
	DLinear	0.8150	1.4997
	NLinear	0.8194	1.5133
	PatchTST	0.3686	0.7565
	STA-Net	0.3874	0.7536
	GSTA-Net	<b>0.3275</b>	<b>0.6589</b>
75%	Linear	1.4608	2.2011
	DLinear	1.4571	2.1893
	NLinear	1.4590	2.1935
	PatchTST	0.7950	1.3386
	STA-Net	0.8819	1.3496
	GSTA-Net	<b>0.6522</b>	<b>1.0573</b>

**Table 2.** Results of different models on the QND

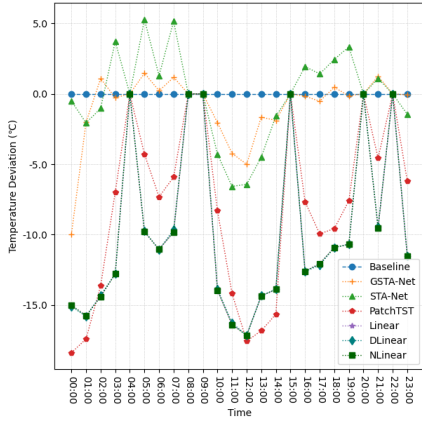
Missing Rate	Methods	MAE ( $\downarrow$ )	RMSE ( $\downarrow$ )
25%	Linear	0.3326	0.8705
	DLinear	0.3359	0.8808
	NLinear	0.3310	0.8682
	PatchTST	0.1426	0.4153
	STA-Net	0.1427	0.4065
	GSTA-Net	<b>0.1340</b>	<b>0.3844</b>
50%	Linear	0.8219	1.5126
	DLinear	0.8184	1.5072
	NLinear	0.8181	1.5073
	PatchTST	0.3569	0.7286
	STA-Net	0.3303	0.6574
	GSTA-Net	<b>0.3230</b>	<b>0.6431</b>
75%	Linear	1.4588	2.1919
	DLinear	1.4759	2.2211
	NLinear	1.4654	2.2069
	PatchTST	0.7628	1.2680
	STA-Net	0.6480	1.0326
	GSTA-Net	<b>0.5993</b>	<b>0.9622</b>



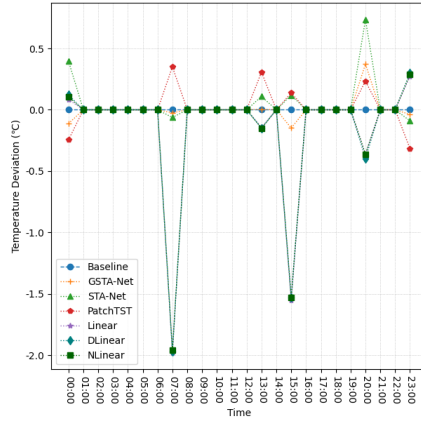
(a)



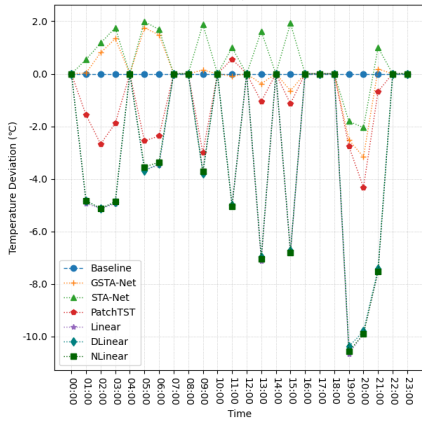
(b)



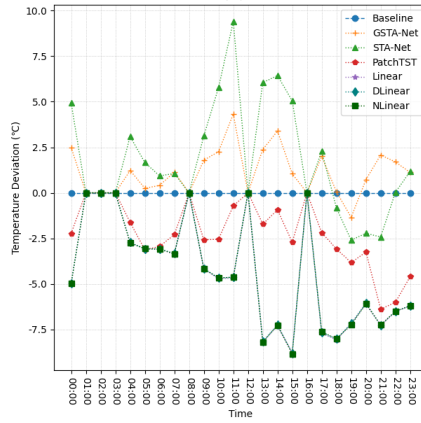
(c)



(d)



(e)



(f)

**Fig. 5.** Show the differences between reconstruction values  $Y$  and real values  $X$  under three different missing rates, where figure (a), (b) and (c) are the reconstruction results of different models at Lhasa meteorological station with missing rates of 25%, 50%, and 75%, respectively; figure (d), (e) and (f) are the reconstruction results of different models at Qamdo meteorological station with missing rates of 25%, 50%, and 75%, respectively.

For better comparison, this study selected five up-to-date models: STA-Net [10], PatchTST [18], DLinear [20], NLinear [20], and Linear [20].

Due to the phenomenon of significant missing data in the meteorological field frequently occurred, this study selected to conduct comparative experiments under the conditions of missing rates of 25%, 50%, and 75%. The experimental results are shown in the table 1 and table 2. And table 1 and table 2 list the performance comparison of GSTA-Net with other methods on the TND and QND, respectively, where the best results are shown in bold. To further demonstrate the performance of GSTA-Net, we have conducted experiments on two real meteorological stations which are Lhasa and Qamdo meteorological station, respectively. The results are shown in the Fig. 5, which shows the difference between the missing filling values and the true values of different models under three different missing rates. Apparently, the filling performance of GSTA-Net is superior to other models. The following will provide a detailed analysis.

According to table 1, table 2 and Fig. 5, it can be inferred that Linear, DLinear, and NLinear exhibit the worst reconstruction performance in experiments, but the filling abilities of each other are commensurate. This may be because of they have similar model architectures, which are adapted to multivariate long sequence data, not univariate short sequence data in paper. Besides, they do not actively utilize the spatiotemporal information in the data, only passively utilize it. Meanwhile, it is also observed that PatchTST and STA-Net have better filling performance than linear models, but inferior to GSTA-Net. This may be due to they only utilize the local spatiotemporal relationships, not the global spatiotemporal information. It is worth noting that the imputing performance of STA-Net is slightly better than that of PatchTST on the dataset QND, which may be derived from that compared to the TND dataset, the QND dataset has a denser distribution of meteorological stations and a more concentrated regional distribution. In the case, it is more conducive to the local spatial attention of STA-Net rather than the patch mechanism of PatchTST. However, since GSTA-Net can fully utilize spatiotemporal information through GSA-Net and GTA-Net, these models cannot compare to it in terms of filling accuracy.

**Ablation experiment.** To verify the effectiveness of GSSA and GTSA, we selected FEM as baseline and carried out corresponding ablation experiments on the TND dataset under missing rates of 25%, 50%, and 75%, respectively. The results are shown in table 3. It can be observed that FEM has the worst filling performance, which is likely due to its insufficient acquisition of the global spatiotemporal information contained in the data. When FEM is combined with GSSA or GTSA, it can utilize global spatial or temporal information, resulting in better filling performance

than FEM. When GSSA and GTSA are integrated into FEM, the best filling performance is achieved due to utilizing global spatiotemporal information.

**Table 3.** Results of ablation experiment

Missing Rate	Methods	MAE ( $\downarrow$ )	RMSE ( $\downarrow$ )
25%	FEM	0.1487	0.4216
	FEM+GSSA	0.1366	0.3935
	FEM+GTSA	0.1361	0.3920
	FEM+GSSA+GTSA	<b>0.1361</b>	<b>0.3913</b>
50%	FEM	0.4154	0.7989
	FEM+GSSA	0.3489	0.6932
	FEM+GTSA	0.3527	0.6985
	FEM+GSSA+GTSA	<b>0.3275</b>	<b>0.6589</b>
75%	FEM	0.9363	1.4252
	FEM+GSSA	0.6979	1.1135
	FEM+GTSA	0.7602	1.1932
	FEM+GSSA+GTSA	<b>0.6522</b>	<b>1.0573</b>

**Table 4.** Comparison of efficiency between GSTA-Net and STA-Net

Missing Rate	Model	Parameter (M)	Time(s/epoch)
25%	STA-Net	192.99	102.62
	GSTA-Net	12.33	35.76
50%	STA-Net	192.49	87.70
	GSTA-Net	12.50	42.34
75%	STA-Net	192.47	84.91
	GSTA-Net	13.23	61.58

**Performance comparison.** Because of the GSTA-Net is proposed based on the STA-Net, and the previous experiments have demonstrated that GSTA-Net has better filling performance compared to STA-Net. This paragraph will demonstrate that GSTA-Net has better operational efficiency compared to STA-Net. Table 4 shows the number of parameters and training time per epoch required for obtaining the optimal model on TND datasets. Obviously, GSTA-Net has a faster convergence speed and fewer parameters compared to STA-Net. Therefore, GSTA-Net has better filling performance compared to STA-Net.

## 5. Conclusion and future work

This paper introduces a novel Global Spatiotemporal Attention Network model (GSTA-Net) for imputing missing temperature data in multiple meteorological stations, by developing three key components: a feature expression model, a global spatial attention mechanism, and a global temporal attention mechanism. Extensive experimental comparisons demonstrate that GSTA-Net can effectively capture the global spatiotemporal information and sustains a comparatively excellent performance in imputing missing values, even at high missing rates. Ablation studies corroborate that both attention mechanisms uniquely enhance the effectiveness of missing value reconstruction.

The model proposed in this paper shows promise for reconstructing missing temperature data at weather stations and offering theoretical grounding and reference for related missing value imputing tasks. However, the GSTA-Net model still has some limitations: It mainly concentrates on missing temperature values without addressing other meteorological data such as wind speed and precipitation, which indicates a potential deficiency in generalization capability. Future work should focus on enhancing the dataset, developing more appropriate network architectures and refining loss functions to boost the model generalization. This improvement will facilitate broader application including support for diverse meteorological data types and expansion into other domains such as sensor, electrical, and similar data.

**Acknowledgments.** This work was supported by the National Natural Science Foundation of Qinghai Province (No.2023-ZJ-906M) and the National Natural Science Foundation of China (No.62162053, No. 42265010, No. 62062059).

**Disclosure of Interests.** None

## References

1. Lompar, M., Lalić, B., Dekić, L., Petrić, M.: Filling gaps in hourly air temperature data using debiased ERA5 data. *Atmosphere* vol. 10(1), pp. 13. Publisher, MDPI (2019)
2. Lara-Estrada, L., Rasche, L., Sucar, L.E., Schneider, U.A.: Inferring missing climate data for agricultural planning using Bayesian networks. *Land* vol. 7(1), pp. 4. Publisher, MDPI (2018)
3. Henn, B., Raleigh, M.S., Fisher, A., Lundquist, J.D.: A comparison of methods for filling gaps in hourly near-surface air temperature data. *Journal of Hydrometeorology* vol. 14(3), pp. 929–945. Publisher, American Meteorological Society (2013)
4. Afrifa-Yamoah, E., Mueller, U.A., Taylor, S., Fisher, A.: Missing data imputation of high-resolution temporal climate time series data. *Meteorological Applications* vol. 27(1), pp. 1873. Publisher, Wiley Online Library (2020)
5. Firat, M., Dikbas, F., Koc, A.C., et al.: Analysis of temperature series: estimation of missing data and homogeneity test. *Meteorological Applications* vol. 19(4), pp. 397–406. Publisher, Wiley Online Library (2012)
6. Kanda, N., Negi, H.S., Rishi, M.S., et al.: Performance of various techniques in estimating missing climatological data over snowbound mountainous areas of Karakoram Himalaya.

- Meteorological Applications vol. 25(3), pp. 337–349. Publisher, Wiley Online Library (2018)
7. Basakin, E.E., Ekmekcioǒlu, Ô, Ôzger, M.: Providing a comprehensive understanding of missing data imputation processes in evapotranspiration-related research: a systematic literature review. *Hydrological Sciences Journal* vol. 68(14), pp. 2089–2104. Publisher, Taylor & Francis (2023)
  8. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* vol. 521(7553), pp. 436–444. Publisher, Springer Nature (2015)
  9. Park, J., Mûller, J., Arora, B., et al.: Long-term missing value imputation for time series data using deep neural networks. *Neural Computing and Applications* vol. 35(12), pp. 9071–9091. Publisher, Springer (2023)
  10. Hou, T., Wu, L., Zhang, X., et al.: STA-Net: Reconstruct Missing Temperature Data of Meteorological Stations Using a Spatiotemporal Attention Neural Network. In: *Proceedings of the Springer Conference on Neural Information Processing*, pp. 29–52 (2023)
  11. Park, J., Yoon, D., Seol, S.J., et al.: Reconstruction of seismic field data with convolutional U-Net considering the optimal training input data. In: *Proceedings of the SEG Conference on International Exposition and Annual Meeting*, pp. D023S027R005 (2019)
  12. Daly, S., Davis, R., Ochs, E., Pangburn, T.: An approach to spatially distributed snow modelling of the Sacramento and San Joaquin basins, California. *Hydrological Processes* vol. 14(18), pp. 3257–3271. Publisher, Wiley Online Library (2000)
  13. Khan, M., Almazah, M.M.A., Eilahi, A., et al.: Spatial interpolation of water quality index based on Ordinary kriging and Universal kriging. *Geomatics, Natural Hazards and Risk* vol. 14(1), pp. 2190853 (2023)
  14. Pape, R., Wundram, D., L’offler, J.: Modelling near-surface temperature conditions in high mountain environments: an appraisal. *Climate Research* vol. 39(2), pp. 99–109 (2009)
  15. Hasanpour, M., Dinpashoh, Y.: Evaluation of efficiency of different estimation methods for missing climatological data. *Stochastic environmental research and risk assessment* vol. 26, pp. 59–71. Publisher, Springer (2012)
  16. Wang, H., Yuan, Z., Chen, Y., Shen, B., Wu, A.: An industrial missing values processing method based on generating model. *Computer Networks* vol. 158, pp. 61–68 (2019)
  17. Chuanjie, X., Chong, H., Deqiang, Z., Wei, H.: BiLSTM-I: A deep learning-based long interval gap-filling method for meteorological observation data. *International Journal of Environmental Research and Public Health* vol. 18(19), pp. 10321. Publisher, MDPI (2021)
  18. Yuqi, N., Nam H, N., Phanwadee, S., Jayant, K.: A time series is worth 64 words: Long-term forecasting with transformers. In: *Proceedings of the International Conference on Learning Representations*, (2023)
  19. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: *Advances in neural information processing systems* vol. 30 (2017)
  20. Ailing, Z., Muxi, C., Lei, Z., Qiang X.: Are transformers effective for time series forecasting. In: *Proceedings of the AAAI Conference on artificial intelligence*, pp. 11121–11128 (2023)
  21. Ahmed, S.F., Alam, M.S.B., Hassan, M., et al.: Deep learning modelling techniques: current progress, applications, advantages, and challenges. *Artificial Intelligence Review* vol. 56, pp. 13521–13617. Publisher, Springer (2023)
  22. Wentao, C., Liang, B.: Contrastive learning with the feature reconstruction amplifier. In: *Proceedings of the AAAI Conference on artificial intelligence*, vol. 37(6), pp. 7279–7287. (2023)
  23. Liu, Z., Yin, C., Li, T.: Chinese Spelling Check Based on BERT and Multi-feature Fusion Embedding. *Computer Science* vol. 50(3), pp. 282–290 (2023)



24. Liu, G., Reda, F.A., Shih, K.J., Wang, T.-C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 85–100 (2018)
25. Tobler, W.R.: A computer movie simulating urban growth in the Detroit region. *Economic geography* vol. 46(sup1), pp. 234–240. Publisher, Clark University (1970)
26. Liu, H., Ruan, Z., Zhao, P., Dong, C., Shang, F., Liu, Y., Yang, L., Timofte, R.: Video super-resolution based on deep learning: a comprehensive survey. *Artificial Intelligence Review* vol. 55(8), pp. 5981–6035. Publisher, Springer (2022)
27. Song, Q., Li, J., Guo, H., et al.: Denoised non-local neural network for semantic segmentation. *IEEE Transactions on Neural Networks and Learning Systems* (2023)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778. Publisher, IEEE (2016)
29. Li, W., Guo, Y., Wang, B., et al.: Learning spatiotemporal embedding with gated convolutional recurrent networks for translation initiation site prediction. *Pattern Recognition* vol. 136, pp. 109234. Publisher, Elsevier (2023)