Global-Local Feature Learning via Dynamic Spatial-Temporal Graph Neural Network in Meteorological Prediction

Yibi Chen, Kenli Li, *Senior Member, IEEE*, Chai Kiat Yeo, and Keqin Li, *Fellow, IEEE*

***Abstract*—The meteorological environment has a profound impact on global health (e.g., air quality), science and technology (e.g., rocket launches), and economic development (e.g., poverty reduction) etc. Meteorological prediction presents numerous challenges to both academia and industry due to its multifaceted nature which encompasses real-time observations and complex modeling. Recent research adopt graph convolutional recurrent network and establish coordinate information to obtain local spatial-temporal pattern. However, the model only utilizes the local spatial-temporal information and fail to fully consider the dynamic meteorological situation. To address the above limitations, we propose a Dynamic Spatial-Temporal Graph Neural Network (DSTGNN) to learn global-local meteorological features. Specifically, we divide the global spatial-temporal information along the timeline to obtain local spatial-temporal information. For the global aspect, we design a random throwedge module during the neighborhood propagation process in graph neural network (GNN) to extract the features and adapt to the dynamic situation. We also establish convolution operation module to learn the features. Next, we perform information fusion on the two modules to capture sufficient features. In addition, we employ graph ordinary differential equation (ODE) network and utilize the coordinate information to obtain the long-term features and coordinate relationships. In the local aspect, we first construct a GNN to conduct graph embedding. Then, we integrate another GNN into a gated recurrent unit (GRU) and also use the coordinate information to explore the features and coordinate relationships. Finally, we combine the global and local features via a global-local features learning layer for meteorological prediction. Experimental results on the four real-world meteorological datasets show that DSTGNN outperforms the baseline models.**

***Index Terms*—Global-local meteorological features, graph ODE network, dynamic spatial-temporal graph neural network, meteorological prediction, random throw edge.**

Manuscript received 8 August 2023; revised 4 March 2024; accepted 21 April 2024. Date of publication 10 May 2024; date of current version 27 September 2024. This work was supported in part by the Key Program of the National Natural Science Foundation of China under Grant U21A20461, and in part by the National Natural Science Foundation of China under Grant 62202149. Recommended for acceptance by J. Tang. *(Corresponding author: Kenli Li.)*

Yibi Chen is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: chenyibi@hnu.edu.cn).

Kenli Li is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: lkl@hnu.edu.cn).

Chai Kiat Yeo is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: asckyeo@ntu.edu.sg).

Keqin Li is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China, and also with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TKDE.2024.3397840

I. Introduction

CURRENTLY, numerous nations are increasingly focusing on the advancement of science and technology, leading to extensive exploitation and utilization of natural resources. This has resulted in significant changes in the meteorological systems, including sea-level rise in cities like Los Angeles, Melbourne, and Shanghai and increasingly extreme weather phenomena [1]. Hurricane Lan devastated parts of Florida in September 2022 causing $113.1 billion in damage and killing at least 160 people. In the 27th United Nations Climate Change Conference in 2022, United Nations Secretary-General Guterres stated global warming is imminent. Establishing an intelligent meteorological system will enhance various meteorological aspects, such as climate and energy through processes like data collection, data mining and technology development. Meteorological prediction plays a vital role within the intelligent meteorological system as it enables early detection and warning of severe weather events. This proactive approach ensures the safety of people, their property, and the environment, with particular relevance to smart cities.

With the rapid development of deep learning, significant advancements have been achieved in the spatial-temporal research on images and videos. Moreover, deep learning technology exemplified by convolutional neural network (CNN) is very efficient in extracting spatial pixel features in multiple temporal dimensions to predict the future. Klein et al. [2] have proposed a dynamic convolutional structure to jointly extract the existing and previous feature maps for short-term weather forecasting. STAM [3] builds a 3D CNN and attention mechanism to exploit global features for spatial-temporal videos. Mehrkanoon [4] have established a data-driven module and CNN to explore multivariate weather data. There are existing literatures which have proposed a number of CNN based spatial-temporal forecasting methods. A shortcoming of these methods is that it is very expensive to extract high-quality data from the massive collection of images and videos.

Given the irregular characteristics of the graph-structure, it can thus be very flexible in representing a rich trove of information [5], [6], [7]. Graph spatial-temporal data is a special graph structure data [8], [9], which can construct the relationships among multiple information. The advent of GNN methods [10], [11] has revitalized the spatial-temporal graph, bringing it renewed vitality. DCRNN [12] proposes bidirectional random

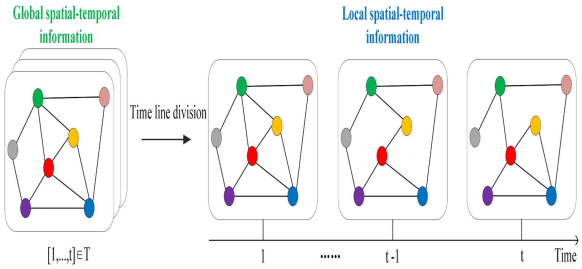


Fig. 1. The relationship of global and local spatial-temporal information.

walks and encoder-decoder structure for spatial and temporal relationships. AGCRN [13] constructs an adaptive mechanism in GCN to capture spatial-temporal features. CLCRN [14] adopts an encoder-decoder structure based on a graph convolutional recurrent network and coordinate information for meteorological forecasting. Although these methods yield good results, they only utilize local spatial-temporal information and do not fully consider the dynamic situation, especially for complex meteorological environments.

To overcome the above challenges, we design a dynamic spatial-temporal graph neural network called DSTGNN, to learn the global-local meteorological features. Fig. 1 shows the global spatial-temporal information being divided along the timeline to obtain multiple sets of local spatial-temporal information as well as the relationship between the global and local information. Compared to local spatial-temporal method, global-local spatial-temporal method explores the interrelated and complementary features. A variety of relationships among nodes can clearly describe the inner nature of things. Spatial-temporal diversity graph enables the model to learn the spatial-temporal diversity of meteorological data. For the global aspect, we construct a random throwedge module during the neighborhood propagation process in GNN to learn the features and consider the dynamic meteorological situation. We also establish convolution operation module to learn the features. Next, we perform information fusion on the two modules to capture sufficient features. Moreover, we construct graph ordinary differential equation (ODE) network and utilize the coordinate information to capture the long-term features and coordinate relationships. For the local aspect, we first construct a GNN for graph embedding. Then, another GNN is integrated into a gated recurrent unit (GRU) which also uses the coordinate information to explore the features as well as the coordinate relationships. Finally, we combine the global and local features by a global-local features learning layer for meteorological prediction. The proposed DSTGNN outperforms existing baseline methods on four real-world meteorological datasets.

The contributions of this paper are summarized as follows:

- We present a DSTGNN framework for meteorological prediction. The model constructs multiple modules (i.e., throwedge, GNN, graph ODE network and GRU) to learn global-local spatial-temporal features.
- We incorporate a random throwedge module during the neighborhood propagation process in GNN to learn the meteorological features and adapt to the dynamic situation.
- We design a graph ODE network to use the coordinate information to obtain the long-term spatial-temporal features as well as the coordinate relationships.
- Experimental results show that the proposed DSTGNN outperforms the baselines, validating the ability of DSTGNN in improving meteorological prediction through global-local feature learning.

II. Related Work

Spatial-temporal data have multiple characteristics (e.g., dynamic and real-time), thus the spatial-temporal forecasting has attracted the interest of researchers, especially for meteorological forecasting in intelligent meteorological system. In this section, we review prior literatures of two categories of prediction techniques, namely, CNN-based spatial-temporal prediction and GNN-based spatial-temporal prediction.

A. CNN-Based Spatial-Temporal Prediction

Many studies have applied CNN-based techniques to complex spatial-temporal prediction. Klein et al. [2] have proposed a dynamic convolutional structure to jointly extract the existing and previous feature maps for short-term weather forecasting. Grover et al. [15] have designed generative and discriminative modules combined with data-driven function to capture the weather information. Mehrkanoon [4] have established a data-driven module and CNN to explore multivariate weather data. SA-ConvLSTM [16] integrates ConvLSTM and self-attention to capture long-term spatial-temporal information. ST-ResNet [17] designs a 2D CNN and a residual structure to learn the spatial-temporal data of multiple regions. LMST3D-ResNet [18] jointly constructs a 3D CNN and resnet to deeply explore the spatial-temporal relationships among multiple local regions. STAM [3] builds a 3D CNN and an attention mechanism to exploit the global features for spatial-temporal videos. Terren-Serrano et al. [19] have adopted a multi-task recurrent neural network to learn dynamic cloud information. Meng et al. [20] have constructed a generative adversarial network, a VGG network and a ConvLSTM to predict sea surface temperature. However, CNN-based spatial-temporal prediction models only use 2D and 3D convolution to learn regularly arranged data such as images and videos.

B. GNN-Based Spatial-Temporal Forecasting

Recently, GNN-based spatial-temporal architectures can efficiently handle irregularly arranged data such as graph structure data. DCRNN [12] proposes bidirectional random walks and an encoder-decoder structure for spatial and temporal relationship, respectively. TGCN [21] builds GCN and GRU to obtain dynamic spatial-temporal dependence. STSGCN [22] designs a synchronous GCN module to learn local spatial-temporal features of different time periods. AGCRN [13] constructs an adaptive mechanism (i.e., node adaptive parameter and data adaptive graph generation) in GCN to capture the spatial-temporal information. STGCN [23] proposes a GCN with multi-scale spatial-temporal structure for middle-long term forecasting.

SSH-GNN [24] proposes a hierarchical GNN based on semi-supervised learning for air quality forecasting. AARGNN [25] integrates GNN with long short-term memory (LSTM) to learn the multiple dynamic features in traffic prediction. GST-PRN [26] establishes a position convolution component and an approximate personalized propagation to improve the spatial dependence relationship and neighborhood information, respectively. STGODE [27] adopts multiple TCN modules and an ODE to capture the long-term traffic spatial-temporal information. AutoSTS [28] presents a GNN-based model to obtain the local spatial-temporal information, and built a CNN-based module to extract the temporal dependent features from multiple ranges. CLCRN [14] adopts an encoder-decoder structure based on a graph convolutional recurrent network and coordinate information for meteorological forecasting. However, it only utilized local meteorological information and did not fully consider the dynamic situation.

Compared to the aforementioned related work, the proposed DSTGNN jointly models the global-local spatial-temporal features and fully considers the dynamic situation in meteorological prediction. Furthermore, DSTGNN is effective as it combines multiple components, namely, throwedge, GNN, graph ODE network and GRU, to explore more spatial-temporal features in depth.

III. Preliminaries and Problem Definition

We introduce some fundamental concepts and problem definition pertaining to graph spatial-temporal meteorological prediction.

Concept 1: Graph Meteorological Network. The graph meteorological network is a directed graph network as $G = (V, E, A)$, where V represents the node set, which is $V = [x_k^M = (x_{k,1}, x_{k,2}, x_{k,3}) \in M^2 : k = 1, 2, \ldots, K]$ to record the coordinate information for K nodes. At time t, K correlated signals are located on the sphere manifold M^2 and $||x_k^M||_2 = 1$ is requested. The expression for the nodes coordinate information adopts the euclidean distance x^D and the sphere x^M. E takes all edges as a set, and $A \in \mathbb{R}^{K\times K}$ denotes the adjacency matrix which is obtained from CLCRN [14].

Concept 2: Graph Meteorological Feature Matrix. The features of each node at time t is expressed as follows:

$$X^{(t)} = [x_1^{(t)}, \ldots, x_{K-1}^{(t)}, x_K^{(t)}] \in \mathbb{R}^{K\times F}. \tag{1}$$

The graph meteorological feature matrix indicates all the nodes' information at various time points as follows:

$$X = [x_1, \ldots, x_{K-1}, x_K]^T \in \mathbb{R}^{K\times F\times T}, \tag{2}$$

where T is the entire temporal period in the historical data and F is the feature dimension of each node.

Concept 3: Coordinate Information. The irregular spatial distribution of the coordinate information used in CLCRN [14] is:

$$\chi(x_j^{i'}; x_i) = \frac{\psi_j^{i'}}{2\pi} \exp(-\frac{(\rho_j^{i'})^2}{\tau}) MLP(|x_j^{i'}, x_i|). \tag{3}$$

where a node j is within the neighborhood of a node i and the angle bisector of pair (i, j) is computed, $\psi_j^{i'} = \arctan(z_j^{i'})$; the angle scale denotes $\psi_j^{i'}/2\pi$, $\rho_j^{i'} = \sqrt{(z_j^{i'})^2 + (\phi_j^{i'})^2}$ while the distance scale represents $\exp(-(\rho_j^{i'})^2/\tau$, where τ is a trainable parameter. The approximator $MLP(\cdot)$ combines smooth activate function (e.g., tanh) to unify the criteria in selection of orthogonal basis.

Problem Definition. The graph meteorological feature matrix on graph meteorological network G, predicting meteorological status of future ξ steps is showed as: $X^{(t+1)}, \ldots, X^{(t+\xi)}$. In meteorological forecasting, we aim to learn the historical observation information through a function P_θ to predict the meteorological conditions in the future ξ steps:

$$\{X, G\} \xrightarrow{P_\theta} (X^{(t+1)}, \ldots, X^{(t+\xi)}; G). \tag{4}$$

IV. Framework

In meteorological prediction, CLCRN [14] employs an encoder-decoder structure based on a graph convolutional recurrent network and coordinate information. The method achieves good results in learning the local spatial-temporal information. This inspires us to also consider the global spatial-temporal information and the dynamic situation. The importance of the global spatial-temporal information is for the predictor to learn the changing trends and overall interactions in meteorology. Furthermore, the importance of considering the dynamic conditions brings us closer to the real-world meteorological environment, particularly in terms of real-time monitoring and variability. Hence the proposed DSTGNN and CLCRN have the following differences: (1) our framework jointly establishes the global-local spatial-temporal information to explore interrelated and complementary features. (2) In the global aspect, we construct random throwedge module during the neighborhood propagation process in GNN to fully consider the dynamic situation; we adopt graph ordinary differential equation (ODE) network and utilize coordinate information to obtain the long-term spatial-temporal features and coordinate relationships. (3) In the local aspect, we do not need to construct an encode-decode structure.

The entire DSTGNN framework is presented in Fig. 2. We input the global spatial-temporal information, i.e., the graph meteorological feature matrix $X \in \mathbb{R}^{K\times F\times T}$, and then the information is divided to two branches. The first branch (i.e., green rectangle) learns the global spatial-temporal features while the second branch (i.e., blue rectangle) obtains multiple local spatial-temporal features $X^{(:t)} \in \mathbb{R}^{K\times F}$ and extracts these features.

In the first branch which learns the global features, we exploit convolution operation for information fusion. We also design a random throwedge module during the neighborhood propagation process in GNN to learn the features and adapt to the dynamic situation. Next, the information fusion combines the output of the GNN and convolution operation to obtain sufficient features. The graph ODE network learns the coordinate information matrix $\chi(x^{i'}) \in \mathbb{R}^{K\times K}$ (i.e., (3)) and the output of the information fusion to explore the long-term correlations of the deep network and the coordinate relationships. Note that the global and local spatial-temporal information in our method is divided along the timeline (i.e., temporal dimension), thus we

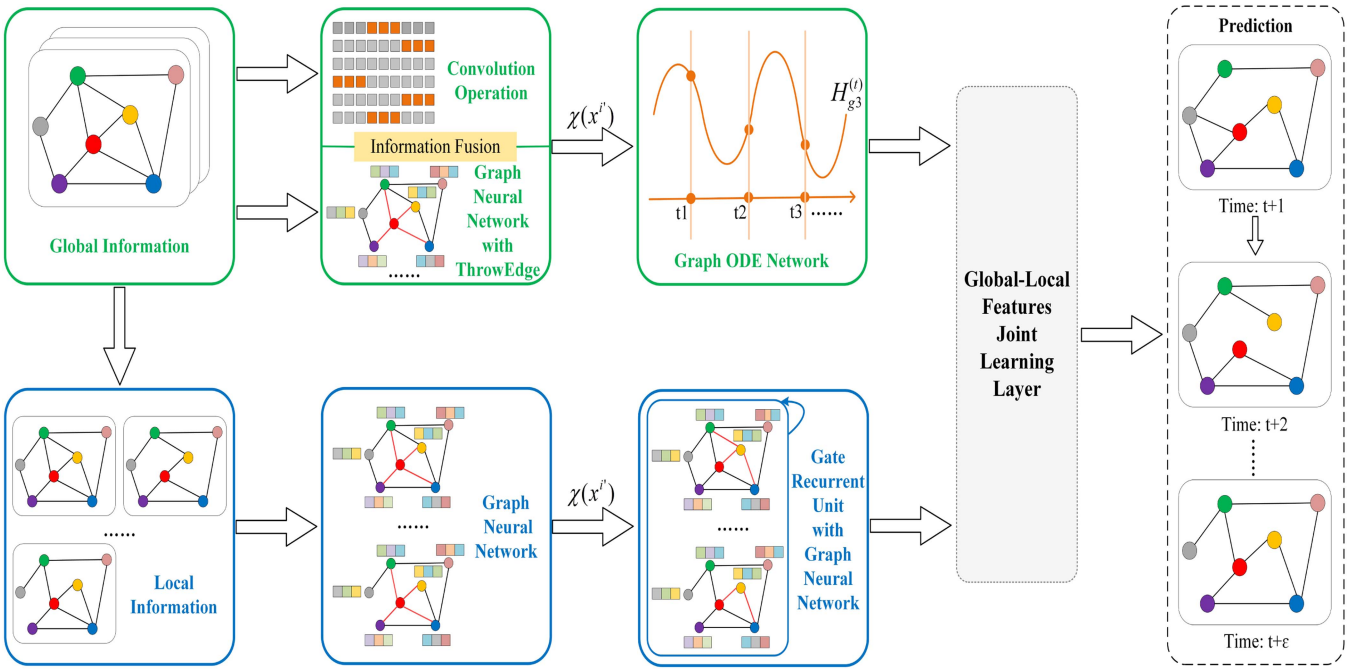


Fig. 2. The overall DSTGNN framework.

can employ the coordinate information (i.e., spatial dimension) to correlate the global and local features.

In the second branch which learns the local features, the local spatial-temporal information is obtained based on the global spatial-temporal information. We first adopt a GNN to perform the graph embedding. Then, we input the coordinate information and graph embedding into the another GNN to learn the feature dependencies and coordinate relationships. The GNN is integrated into a GRU to further exploit the features, especially for the local time series.

In the global-local features learning layer, we aggregate the global features and local features, and then perform a convolution operation to handle these global-local features to generate prediction.

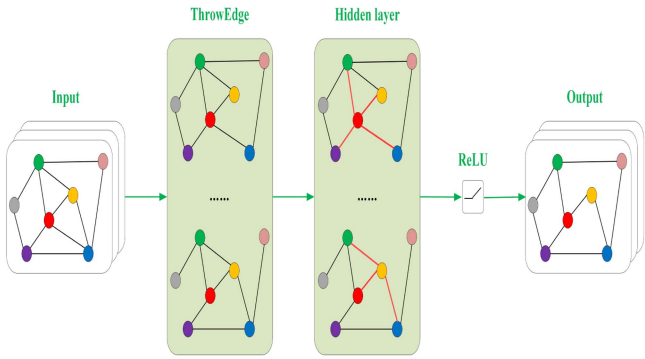


Fig. 3. Graph neural network with throwedge for global spatial-temporal features.

A. Global Spatial-Temporal Information

Global spatial-temporal information can reveal the overall changing trends and interactions, and contribute to better understanding of the meteorological characteristics of an area.

1) Convolution Operation: The purpose of the convolution operation is to build a information fusion. We convert the global spatial-temporal information $X \in \mathbb{R}^{K\times F\times T}$ into $X_1 \in \mathbb{R}^{F\times K\times T}$ which performs the convolution operation to obtain $H_{g1} \in \mathbb{R}^{C\times K\times T}$.

2) Graph Neural Network With ThrowEdge: GNN adopts message passing and information aggregation mechanisms to capture the node features and neighborhood relationships. Meteorological environments are more dynamic and variable in real-time than other types of spatial-temporal features such as traffic conditions, especially when some features can be fleeting and hence are lost at a point in time. Therefore, we design a throwedge module in GNN to fully consider the dynamic situation as shown in Fig. 3. The ThrowEdge module randomly throws a certain proportion of edges. Some non-zero data Q_p in the adjacency matrix A randomly enforces zeros, in which Q represents the number of edges and p denotes the throwing edges rate. The adjacency matrix A_{throw} is given as follows:

$$A_{throw} = A - \hat{A}, \tag{5}$$

where A establishes the relationships of each pair edge (e.g., v_i, v_j) to present A_{ij}. The sparse matrix $\hat{A}$ is a random set of size Q_p extracted from the raw edges E. We apply re-normalization for A_{throw} to obtain $\widetilde{A}_{throw}$.

The global spatial-temporal features $X \in \mathbb{R}^{K\times F\times T}$ are transformed into $X_2 \in \mathbb{R}^{T\times K\times F}$. GNN extracts X_2 to get $H_{g2} \in$

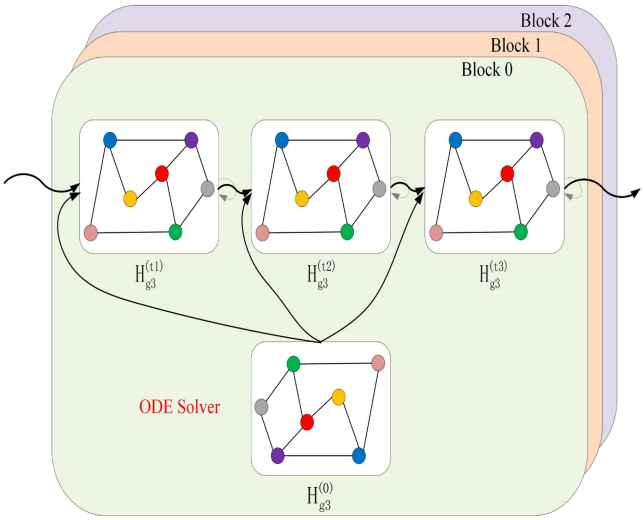


Fig. 4. The chartflow of multi-block ODE solver.

$\mathbb{R}^{T\times K\times C}$ as shown:

$$H_{g2} = ReLU(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}X_2W_2). \tag{6}$$

where $\widetilde{D}$ is the diagonal degree matrix with $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$. The node degree is denoted as $d = \{d_1, \ldots, d_K\}$, in which d_i computes all the edge weights which are related to node i. The diagonal elements of $\widetilde{D}$ are given by d. $\widetilde{A} = A + I_m$ is an adjacency matrix with added self-connection, I_m is the identity matrix. $W_2 \in \mathbb{R}^{S\times F\times C}$ is a transform matrix.

In (6), we replace $\widetilde{A}$ with $\widetilde{A}_{throw}$ for neighborhood propagation, and the GNN with throwedge is expressed as follows:

$$H_{g2} = ReLU(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}_{throw}\widetilde{D}^{-\frac{1}{2}}X_2W_2). \tag{7}$$

Next, we further discuss GNN with throwedge. Throwedge can bring multi-perturbations for the graph connections, generating various random changes to the input data. This process can be regarded as a type of data augmentation mode for the graph representation. For the neighborhood, throwedge randomly deletes the edges in the original graph and then carries out neighbor propagation; it aggregates a random subset rather than all the neighborhood information during GNN training. Thus, the GNN with throwedge can effectively adapt to the dynamic meteorological environment.

3) Graph ODE Network: The graph ordinary differential equation (ODE) network overcomes the limitation of the shallow GNN as the latter is inefficient in capturing long-term spatial-temporal information. This network constructs the continuous dynamic system by parameterizing the derivative of the hidden states. Fig. 4 shows a multi-block ODE solver that describes the current state and the original state to infer the functions in the hidden state. Moreover, we introduce coordinate information in the graph ODE network to explore the spatial coordinate relationships.

As shown in Fig. 2, one input to the graph ODE network $H_{g3}^{\{0\}}$ adopts the output of a convolutional operation and the GNN with throwedge, in which we transform $H_{g1} \in \mathbb{R}^{C\times K\times T}$ to $H'_{g1} \in \mathbb{R}^{K\times T\times C}$ and $H_{g2} \in \mathbb{R}^{T\times K\times C}$ to $H'_{g2} \in \mathbb{R}^{K\times T\times C}$. Then, the convolutional operation H'_{g1} and the GNN with throwedge H'_{g2} are combined by a information fusion. The other input as shown in Fig. 2 is the coordinate information $\chi(x^{i'})$. Next, the discrete version of the ODE mathematical function is denoted as follows:

$$H_{g3}^{\{l+1\}} = H_{g3}^{\{l\}} \times_1 \frac{\alpha}{2}\widetilde{A} \times_2 U \times_3 W' + H_{g3}^{\{0\}}, \tag{8}$$

where $H_{g3}^{\{l\}}$ denotes the spatial-temporal features of the hidden state on the l-th layer; $\alpha \in [0, 1]$ is a hyperparameter; $\times_i$ represents the i mode of matrix multiplication; U indicates the transform matrix in the temporal aspect; W' presents the transform matrix in the feature aspect.

In this way, both the spatial and temporal features can be handled simultaneously. The complex spatial-temporal relationships are coupled based on the tensor multiplication of multiple mode $\times_i$. By expanding (8), the restart distribution $H_{g3}^{\{0\}}$ restrains the over-smoothing problem as follows:

$$H_{g3}^{\{l\}} = \sum_{i=0}^{l}\left(H_{g3}^{\{0\}} \times_1 \frac{\alpha}{2}\widetilde{A}^i \times_2 U^i \times_3 W'^i\right), \tag{9}$$

where $H_{g3}^{\{l\}}$ aggregates the features of all layers while the original features are utilized.

To verify the effective restart distribution, this version does not consider $H_{g3}^{\{0\}}$ as follows:

$$H_{g3}^{\{l+1\}} = H_{g3}^{\{l\}} \times_1 \frac{\alpha}{2}\widetilde{A} \times_2 U \times_3 W', \tag{10}$$

the final form is:

$$H_{g3}^{\{n\}} = H_{g3}^{\{0\}} \times_1 \frac{\alpha}{2}\widetilde{A}^n \times_2 U^n \times_3 W'^n. \tag{11}$$

Suppose the eigenvalue decomposition of $\widetilde{A}$ is $\widetilde{A} = P\Lambda P^T$, and $\Lambda = diag(\lambda_1, \ldots, \lambda_{z-1}, \lambda_z)$ denotes the diagonal matrix:

$$\begin{aligned}\widetilde{A}^n &= Pdiag(\lambda_1, \ldots, \lambda_{z-1}, \lambda_z)P^T \\ &= \lambda_1^n Pdiag\left(1, (\frac{\lambda_2}{\lambda_1})^n, \ldots, (\frac{\lambda_z}{\lambda_1})^n\right)P^T \\ &\Rightarrow \lambda_1^n Pdiag(1, 0, \ldots, 0)P^T,\end{aligned} \tag{12}$$

where n is close to infinity with $\lambda_1 > \lambda_{z-1} > \lambda_z$. Note that the diagonal elements only keep the largest factor while the other elements are all zeros which result in loss of too many features.

The equation is extended to obtain the Riemann sum form from 0 to n on i, and n is replaced with a continuous variable t as follows:

$$\begin{aligned}H_{g3}^{\{n\}} &= \sum_{i=0}^{n}(H_{g3}^{\{0\}} \times_1 \frac{\alpha}{2}\widetilde{A}^i \times_2 U^i \times_3 W'^i) \\ &= \sum_{i=1}^{n+1}(H_{g3}^{\{0\}} \times_1 \frac{\alpha}{2}\widetilde{A}^{(i-1)\times\triangle t} \times_2 \\ &\quad U^{(i-1)\times\triangle t} \times_3 W'^{(i-1)\times\triangle t} \triangle t),\end{aligned} \tag{13}$$

When $t = n$, the condition $\triangle t = \frac{t+1}{n+1}$ is satisfied. However, as n is close to ∞, the equation is thus defined as follows:

$$H_{g3}^{(t)} = \int_0^{t+1} H_{g3}^{\{0\}} \times_1 \frac{\alpha}{2}\widetilde{A}^{\tau} \times_2 U^{\tau} \times_3 W'^{\tau} d\tau \quad (14)$$

Intuitively, we obtain the expression of ODE form as:

$$\frac{dH_{g3}^{(t)}}{dt} = H_{g3}^{\{0\}} \times_1 \frac{\alpha}{2}\widetilde{A}^{t+1} \times_2 U^{t+1} \times_3 W'^{t+1}, \quad (15)$$

where $\widetilde{A}^{t+1}$, U^{t+1}, W'^{t+1} entail complex computation processes when t is not an integer.

For more detailed description and reasoning analysis of the above equations, please refer to STGODE [27]. The expression of ODESolver is presented as follows:

$$H_{g3}^{(t)} = ODESolve\left(\frac{dH_{g3}^{(t)}}{dt}, H_{g3}^{\{0\}}, t\right), \quad (16)$$

where $H_{g3}^{\{0\}}$ presents the initial input, $H_{g3}^{(t)} \in \mathbb{R}^{K\times T\times C}$ is the output of the global spatial-temporal features; $\frac{dH_{g3}^{(t)}}{dt} = H_{g3}^{(t)} \times_1 (\frac{\alpha}{2}\widetilde{A} - I_m) + H_{g3}^{(t)} \times_2 (U - I_m) + H_{g3}^{(t)} \times_3 (W - I_m) + H_{g3}^{\{0\}}$.

To explore the coordinate relationships among the nodes, we replace the adjacency matrix $\widetilde{A} \in \mathbb{R}^{K\times K}$ with the coordinate information matrix $\chi(x^{i'}) \in \mathbb{R}^{K\times K}$; $\frac{dH_{g3}^{(t)}}{dt} = H_{g3}^{(t)} \times_1 (\frac{\alpha}{2}\chi(x^{i'}) - I_m) + H_{g3}^{(t)} \times_2 (U - I_m) + H_{g3}^{(t)} \times_3 (W - I_m) + H_{g3}^{\{0\}}$.

Algorithm 1 summarizes the learning process of the global spatial-temporal information.

B. Local Spatial-Temporal Information

The local spatial-temporal information can specifically reflect the spatial state at a point in time, so extracting the information can enable deep exploration of the range of the meteorological conditions.

The different local spatial-temporal information $X^{(:t)} \in \mathbb{R}^{K\times F}$ is obtained by dividing the global spatial-temporal information $X \in \mathbb{R}^{K\times F\times T}$ along the timeline.

1) Graph Neural Network: GNN carries out graph embedding to obtain $H_{l1}^{(:t)} \in \mathbb{R}^{K\times C}$ as follow:

$$H_{l1}^{(:t)} = ReLU(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}X^{(:t)}W_3), \quad (17)$$

where $W_3 \in \mathbb{R}^{F\times C}$ is a transform matrix.

2) Gate Recurrent Unit With Graph Neural Network: We input $H_{l1}^{(:t)}$ and $\chi(x^{i'})$ into GNN to learn the features and the different node coordinate relationships. The expression of GRU with GNN is shown as:

$$z^{(t)} = \sigma(\chi(x^{i'}))[H_{l1}^{(:t)}, h^{(t-1)}]W_z + b_z),$$
$$u^{(t)} = \sigma(\chi(x^{i'}))[H_{l1}^{(:t)}, h^{(t-1)}]W_u + b_u),$$
$$\hat{H}_{l2}^{(:t)} = tanh(\chi(x^{i'})[H_{l1}^{(:t)}, z^{(t)} \odot h^{(t-1)}]W_{\hat{h}} + b_{\hat{h}}),$$

Algorithm 1: Global Spatial-Temporal Information Algorithm.

Require: The graph meteorological network $G = (V, E, A)$;
Historical meteorological data: X;
Ensure: Learned Global Spatial-Temporal Information.
//Convolution operation;
$X_1 \leftarrow X$;
3: $H_{g1} \leftarrow X_1$
//Graph neural network with throwedge;
$A_{throw} \leftarrow A$;
6: $X_2 \leftarrow X$;
$H_{g2} \leftarrow ReLU(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}_{throw}\widetilde{D}^{-\frac{1}{2}}X_2W_2)$;
//Information fusion;
9: $H'_{g1} \leftarrow H_{g1}$;
$H'_{g2} \leftarrow H_{g2}$;
$H_{g3}^{\{0\}} \leftarrow H'_{g1} + H'_{g2}$;
12: *//Graph ODE network*;
$\chi(x^{i'}) \leftarrow X$;
$(H_{g3}^{\{0\}}, \chi(x^{i'}))$ conveys to $ODEsolver$;
15: **for** all blocks (0,1,2) **do**
$H_{g3}^{(t)} = ODESolve(\frac{dH_{g3}^{(t)}}{dt}, H_{g3}^{\{0\}}, t)$;
end for
18: **return** $H_{g3}^{(t)}$

$$H_{l2}^{(:t)} = u^{(t)} \odot h^{(t-1)} + (1 - u^{(t)}) \odot \hat{H}_{l2}^{(:t)} \quad (18)$$

where $z^{(t)}$ denotes the reset gate and $u^{(t)}$ represents the update gate at t time; σ represents an activate function (i.e., Sigmoid); $h^{(t-1)}$ is a hidden state at time $t-1$; W_z, W_u and $W_{\hat{h}}$ are training parameter matrices; $\odot$ indicates Hadamard Product; $H_{l1}^{(:t)} \in \mathbb{R}^{K\times C}$ is the input; $H_{l2}^{(:t)} \in \mathbb{R}^{K\times C}$ is the output of the local spatial-temporal features, and then we aggregate all the local spatial-temporal features $H_{l2}^{(:t)} \in \mathbb{R}^{K\times C}$ to obtain $H_{l3}^{(t)} \in \mathbb{R}^{K\times T\times C}$.

Algorithm 2 summarizes the learning process of local spatial-temporal information.

C. Global-Local Features Learning Layer

The global spatial-temporal features are processed by a convolution operation, a GNN with throwedge and a graph ODE network to obtain $H_{g3}^{(t)} \in \mathbb{R}^{K\times T\times C}$. The local spatial-temporal features are extracted by the GNN and the GRU with GNN modules to capture $H_{l3}^{(t)} \in \mathbb{R}^{K\times T\times C}$. The joint learning of the global-local spatial-temporal features is expressed as:

$$H = H_{g3}^{(t)} + H_{l3}^{(t)} \quad (19)$$

Finally, we perform a convolution operation on H to generate the prediction.

Algorithm 3 summarizes the proposed DSTGNN model for global-local spatial-temporal features learning.

Algorithm 2: Local Spatial-Temporal Information Algorithm.

Require: The graph meteorological network $G = (V, E, A)$;
Historical meteorological data: X;
Ensure: Learned Local Spatial-Temporal Information.
$D \leftarrow \emptyset$;
for whole time interval $T(1, \ldots, t)$ **do**
//Obtaining local spatial-temporal information;
$X^{(:t)} \leftarrow X$;
//Graph neural network;
$H_{l1}^{(:t)} \leftarrow ReLU(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}X^{(:t)}W_3)$;
//Gate recurrent unit with graph neural network;
$H_{l2}^{(:t)} \leftarrow (H_{l1}^{(:t)}, \chi(x^{i'}))$;
$(H_{l2}^{(:t)})$ conveys to D;
end for
while $D \neq \emptyset$ **do**
Take each time point of the instance in D to aggregate all local spatial-temporal features $H_{l3}^{(t)}$;
end while
return $H_{l3}^{(t)}$

Algorithm 3: DSTGNN Algorithm.

Require: The graph meteorological network $G = (V, E, A)$;
Historical meteorological data: $X \in \mathbb{R}^{K \times F \times T}$;
Ensure: Learned Global-Local Spatial-Temporal Features.
//Training model;
while stopping criteria is not met **do**
Initialize all training parameters θ in DSTGNN;
Put global spatial-temporal information X into the model;
//Learned global spatial-temporal features;
$H_{g3}^{(t)} \leftarrow X$;
//Learned local spatial-temporal features;
for whole time interval $T(1, \ldots, t)$ **do**
$X^{(:t)} \leftarrow X$;
$H_{l1}^{(:t)} \leftarrow X^{(:t)}$;
$H_{l2}^{(:t)} \leftarrow (H_{l1}^{(:t)}, \chi(x^{i'}))$;
$(H_{l2}^{(:t)})$ conveys to D;
end for
while $D \neq \emptyset$ **do**
$H_{l3}^{(t)} \leftarrow D$;
end while
//Learned global-local spatial-temporal features;
$H \leftarrow (H_{g3}^{(t)}, H_{l3}^{(t)})$;
Find the function P_θ;
end while
return

V. Experimental Evaluation

We evaluate the proposed DSTDNN on four real world meteorological datasets as well as against multiple baseline methods through a series of experiments.

A. Dataset

The datasets contain 2048 meteorological nodes which are distributed across the earth, and the data are provided by WeatherBench [29]. We choose four types of meteorological data, namely, temperature, surface wind, cloud cover and humidity, and the units are K, ms^{-1}, $\% \times 10^{-1}$, $\% \times 10^{-1}$ respectively. The data are collected from 1/1/2010 to 12/31/2018.

B. Settings

The performance evaluation uses two common metrics, i.e., Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

$$MAE(P_\theta^{(t,T)}, \bar{P}_\theta^{(t,T)}) = \frac{1}{T}\sum_{n=t}^{T}|\bar{P}_\theta^{(n)} - P_\theta^{(n)}| \tag{20}$$

$$RMSE(P_\theta^{(t,T)}, \bar{P}_\theta^{(t,T)}) = \frac{1}{T}\sqrt{\sum_{n=t}^{T}|\bar{P}_\theta^{(n)} - P_\theta^{(n)}|^2} \tag{21}$$

where $P_\theta^{(t,T)}$ is the ground truth and $\bar{P}_\theta^{(t,T)}$ represents the prediction.

The input is 12 time steps (i.e., 12 hours) of historical observation data and the prediction horizon is 12 time steps (i.e., 12 hours) in the future. The batch size is set to 32. All the methods employ Adam optimizer to train to a maximum of 100 epochs. The hyperparameters are set as follow: the learning rate is 0.005; the hidden dimension of the GNN is 32; the throwedge rate is 0.1; the hidden dimension of multiple blocks on the graph ODE network are 32, 16 and 32. The different random seeds,(2021, 2022, 2023, 2024, 2025) are used for training. Early stopping is employed in the model training and the patience value is set at 30.

C. Baseline Methods

- TGCN [21] employs a GCN and a GRU to capture dynamic spatial-temporal dependence.
- STGCN [23] models a GCN and a multi-scale spatial-temporal structure for middle-long term forecasting.
- MSTGCN [30] utilizes a GCN and convolution operation to capture the spatial-temporal information.
- ASTGCN [30] constructs an attention module and a GCN to learn the spatial-temporal correlations.
- NET^3 [31] uses a GCN and long short-term memory to explore the implicit temporal relationships.
- GCGRU [32] builds a GCN and a recurrent network for graph-level classification and node-level prediction.
- DCRNN [12] designs a bidirectional random walk and an encoder-decoder structure to capture the spatial and temporal relationships.
- AGCRN [13] establishes an adaptive mechanism including node adaptive parameter and data adaptive graph generation in GCN to capture the spatial-temporal features.

TABLE I
METHOD RESULT COMPARISON ON METEOROLOGICAL PREDICTION

Methods	Temperature		Wind		Cloud cover		Humidity	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
TGCN	3.876 ± 0.103	5.871 ± 0.125	4.173 ± 0.033	5.675 ± 0.043	2.398 ± 0.023	3.657 ± 0.015	1.465 ± 0.028	2.096 ± 0.062
STGCN	4.358 ± 1.048	6.866 ± 1.117	3.658 ± 0.001	4.823 ± 0.004	2.021 ± 0.041	2.958 ± 0.063	0.806 ± 0.245	1.122 ± 0.279
MSTGCN	1.233 ± 0.031	1.921 ± 0.005	1.945 ± 0.013	2.913 ± 0.012	1.876 ± 0.003	2.865 ± 0.011	0.613 ± 0.002	0.873 ± 0.013
ASTGCN	1.498 ± 0.021	2.471 ± 0.003	2.101 ± 0.001	3.152 ± 0.001	1.987 ± 0.001	2.949 ± 0.003	0.724 ± 0.025	1.052 ± 0.048
NET^3	1.338 ± 0.016	2.239 ± 0.025	1.458 ± 0.016	2.437 ± 0.015	1.845 ± 0.005	2.823 ± 0.018	0.568 ± 0.028	0.864 ± 0.032
GCGRU	1.331 ± 0.154	2.172 ± 0.193	1.421 ± 0.007	2.300 ± 0.005	1.596 ± 0.000	2.562 ± 0.002	0.512 ± 0.001	0.811 ± 0.004
DCRNN	1.334 ± 0.092	2.193 ± 0.114	1.442 ± 0.001	2.339 ± 0.008	1.615 ± 0.006	2.584 ± 0.005	0.521 ± 0.003	0.813 ± 0.015
AGCRN	1.267 ± 0.012	1.948 ± 0.031	2.422 ± 0.125	3.419 ± 0.117	1.762 ± 0.152	2.763 ± 0.171	0.584 ± 0.170	0.857 ± 0.213
CLCRN	1.184 ± 0.028	1.916 ± 0.172	1.328 ± 0.053	2.135 ± 0.087	1.496 ± 0.004	2.475 ± 0.003	0.457 ± 0.004	0.718 ± 0.015
DSTGNN	$\mathbf{0.866 \pm 0.009}$	$\mathbf{1.448 \pm 0.132}$	$\mathbf{1\,205 \pm 0.011}$	$\mathbf{1.916 \pm 0.023}$	$\mathbf{1.485 \pm 0.016}$	$\mathbf{2.406 \pm 0.005}$	$\mathbf{0.439 \pm 0.006}$	$\mathbf{0.674 \pm 0.001}$

- CLCRN [14] proposes an encoder-decoder structure based on a graph convolutional recurrent network and coordinate information for meteorological forecasting.

D. Results Evaluation and Analysis on Meteorological Prediction

Table I shows the performance of the proposed DSTGNN against the multiple baselines. The results show that the DSTGNN outperforms the SOTA baselines on the four meteorological datasets. From Table I, it can be observed that comprehensive performance of benchmark methods, TGCN, STGCN, MSTGCN and ASTGCN achieve poorer results compared to NET^3, GCGRU, DCRNN, AGCRN and CLCRN, the latter group is based on recurrent networks.

Therefore, we further analyze the recurrent network based baseline methods. NET^3, GCGRU, DCRNN and AGCRN construct GCN-based recurrent networks to effectively capture the spatial-temporal features. However, these methods do not model the coordinate information. CLCRN adopts an encoder-decoder structure, which establishes the coordinate information of the meteorological features, but neither considers the global features nor considers the dynamic situation fully.

The proposed DSTGNN models the global-local features, in which convolution operation, GNN with throwedge and graph ODE network learn the global features; GNN with GRU extracts the local features. The throwedge module fully considers the dynamic meteorological situation. As a result, DSTGNN achieves better prediction results than the SOTA baselines.

E. Results Analysis at Different Steps

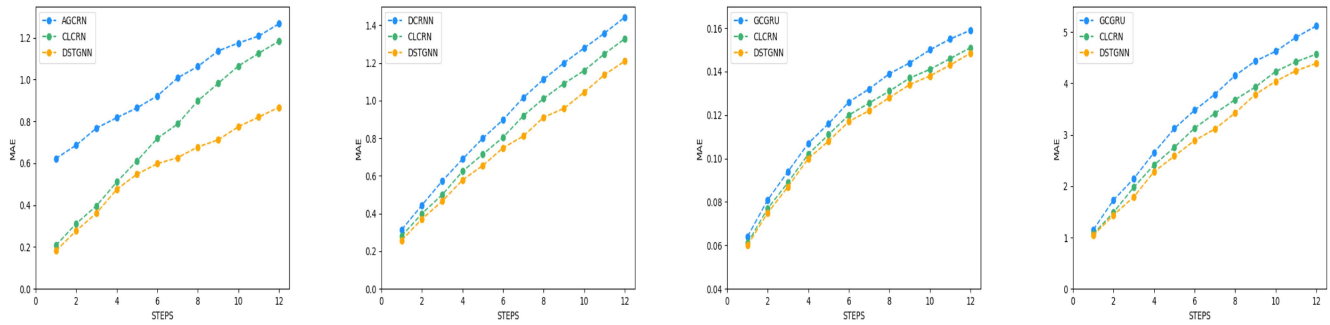
We evaluate the prediction performance of DSTGNN and the best performing two SOTA baselines when the prediction horizon is varied for four meteorological datasets, namely temperature, wind, cloud cover and humidity. Note that 1 time step corresponds to 1 hour. The results are shown in Fig. 5. Generally, as expected the prediction accuracy all degrades as the number of steps or prediction horizon increases.

For the temperature dataset, we observe that the gap between CLCRN and AGCRN decreases as the number of steps increases as opposed to the other datasets where the gap between them widens with the number of steps. On the other hand, DSTGNN does not degrade as fast as them. For the wind dataset, the gap among DSTGNN, CLCRN and DCRNN widens as the number of steps increases. For the cloud cover and humidity datasets, the gap between CLCRN and GCGRU increases with the number of steps while the degradation in DSTGNN's performance tapers off.

In summnary, DSTGNN has been proven to achieve good results with much less performance degradation with increasing prediction horizon compared to the SOTA baselines. This attests to DSTGNN's robustness and effectiveness.

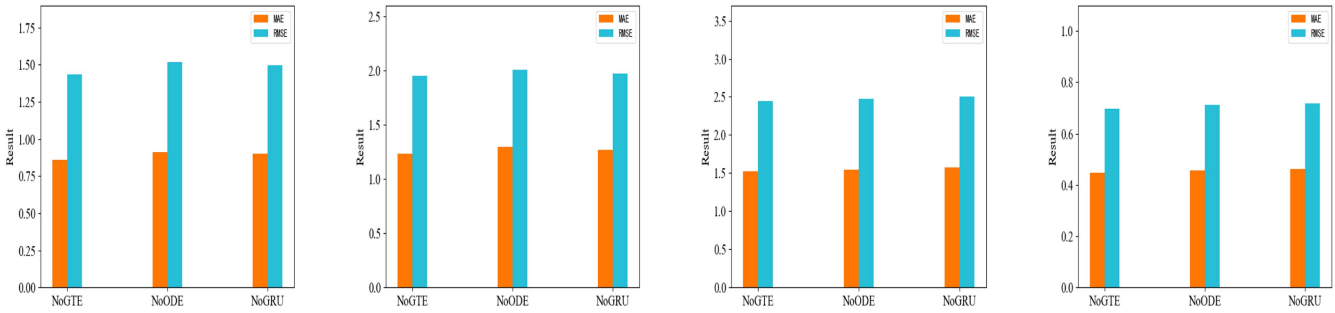
F. Ablation Study

To further demonstrate the performance contribution of the different modules in the proposed DSTGNN, we carry out experiments omitting the different modules as follow:



(a) Temperature (b) Wind (c) Cloud cover (d) Humidity

Fig. 5. MAE results of different steps.



(a) Temperature (b) Wind (c) Cloud cover (d) Humidity

Fig. 6. Performance of different components.

- DSTGNN-noGTE: omits the GNN with throwedge module to study the robustness of the model for dynamic meteorological situation.
- DSTGNN-noODE: omits the graph ODE network module to research the effect of deep networks on long-term prediction.
- DSTGNN-noGRU: removes the GRU module to sutdy the influence of the recurrent neural network.

From the results in Fig. 6(a) and (b), we observe that DSTGNN-noODE has a slightly higher impact on performance than DSTGNN-noGRU and DSTGNN-noGTE, which demonstrates that this module is more sensitive. Furthermore, the small performance change in DSTGNN-noGTE indicates that the model is more robust especially under dynamic meteorological conditions.

From the results in Fig. 6(c) and (d), we observe that DSTGNN-noODE and DSTGNN-noGRU have slightly more effect on performance than DSTGNN-noGTE, which shows that these modules have better learning features ability.

In summary, the different components contribute to the overall performance of DSTGNN, resulting in an effective framework for meteorological forecasting.

G. Performance Analysis for Multiple ThrowEdge Rate

The throwedge module can reflect the dynamic meteorological conditions more realistically, but the loss of some edge information may degrade its performance. As such, we will study the performance impact with different throwedge rates. The different throwedge rates used are 0.1, 0.2 and 0.3.

From Fig. 7, it can be observed that the performance of DSTGNN drops due to loss of edge information when the throwedge rate is higher for the temperature and wind datasets. The same outcome is observed in Fig. 8 for cloud cover and humidity. Overall, the proposed DSTGNN is able to fully consider the dynamic situations and maintains its performance even in the presence of edge information loss.

H. Visualization

Figs. 9 and 10 show the visualization of the DSTGNN prediction results versus ground-truths on the four dataset. We observe that the prediction curve shows the same trend as the ground-truth curve from 0 to 2000 hours.

In Fig. 9(a), the curves show an upward trend from t = 0 to 450 and a downward trend from t = 450 to 900, followed by a sharp rise between t = 900 to 950, a decline from t = 950 to 1250 and thereafer, fluctuations. In Fig. 9(b), the prediction curve is close to the ground-truth curve between t = 0 and 250, slightly higher than the truth curve from t = 250 to 1700 and fluctuations thereafter.

In Fig. 10(a), the curves are sparse between t = 0 and 220, and very sparser from t = 220 to 410, relatively dense from t = 410 to 1100, dense from t = 1100 to 1650, then some sparseness

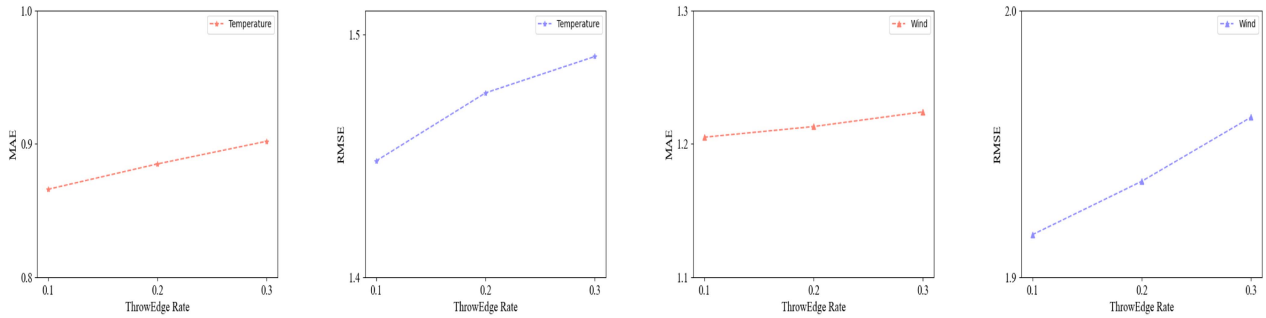


Fig. 7. The performance of different throwedge rate on temperature and wind.

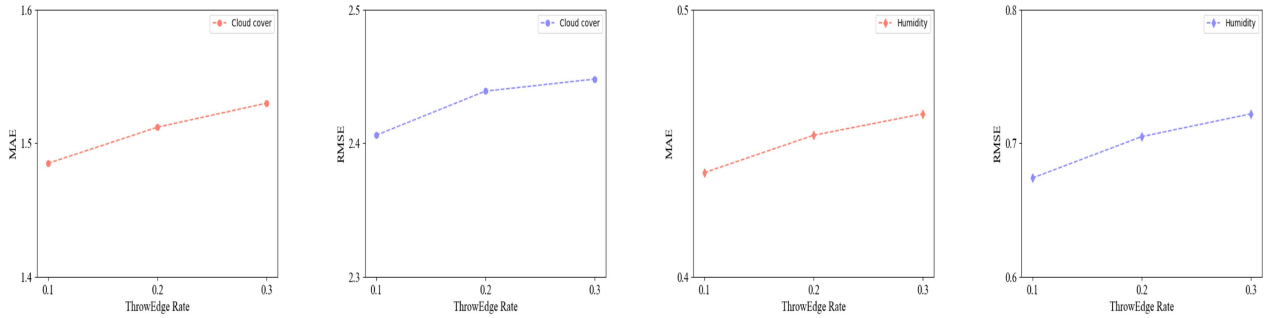
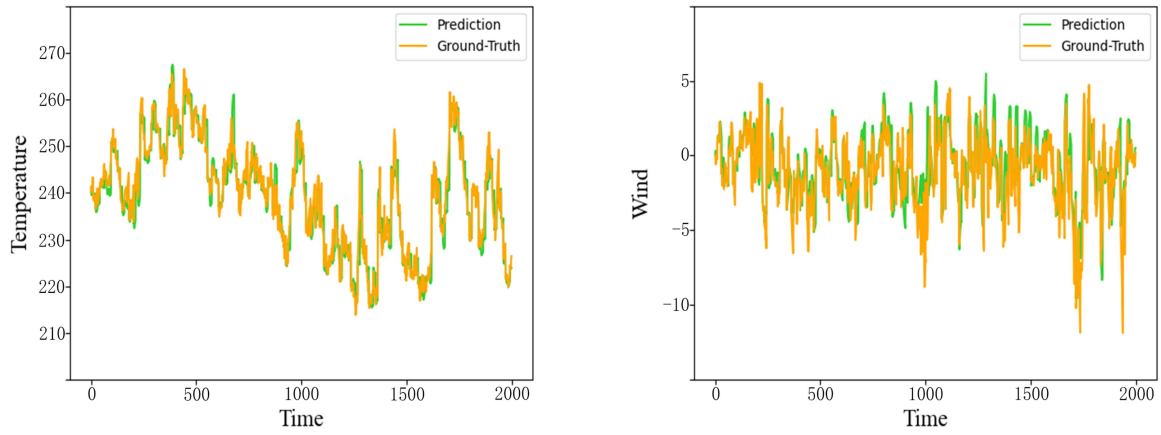


Fig. 8. The performance of different throwedge rate on cloud cover and humidity.



(a) Temperature (b) Wind

Fig. 9. Visualization of ground-truth and prediction on temperature and Wind.

from t = 1650 to 1800 and relatively dense from t = 1800 to 2000. In Fig. 10(b), the curves are relatively stable from t= 0 to 400 with obvious change from t = 400 to 1400, and some change from t = 1400 to 2000.

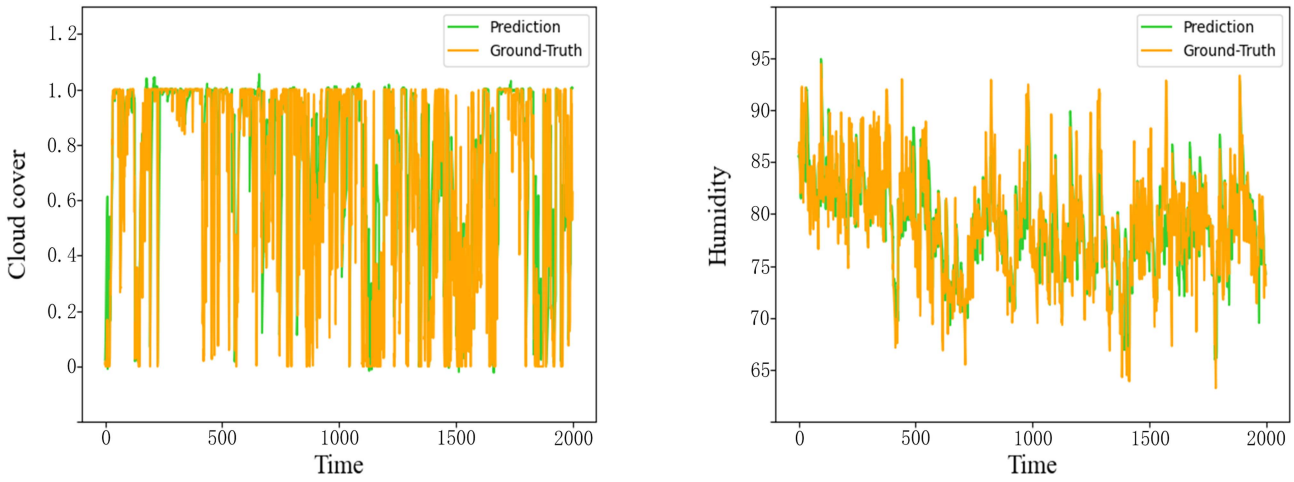
In summary, Figs. 9 and 10 shows that the predictions closely track the ground-truths, testifying to that our model is effective in predicting future meteorological situation.

I. Results Analysis on Various Epochs

We evaluate DSTGNN using different epochs. From Tables II and III, we can observe that the results improve with increasing number of epochs. Table II shows that the model registers stable changes between 60 and 80 and converges at epoch 90 on the temperature dataset and between 60 and 90 on the wind dataset. Table III shows that the model registers a steady improvement on the cloud cover and the humidity datasets across the increasing epoch number and converges at epoch 90.

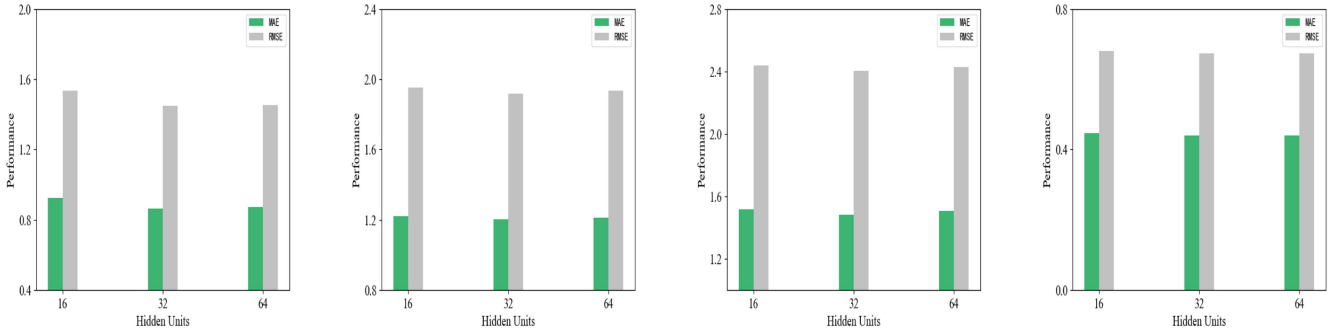
J. Sensitivity Analysis

We vary the number of hidden units to evaluate the performance of the proposed DSTGNN on the different datasets for sensitivity analysis. From Fig. 11, we can observe that hidden unit 16 performs worse compared to hidden units 32, 64. For



(a) Cloud cover (b) Humidity

Fig. 10. Visualization of ground-truth and prediction on cloud cover and humidity.



(a) Temperature (b) Wind (c) Cloud cover (d) Humidity

Fig. 11. Sensitivity of hidden units on different dataset.

TABLE II
THE RESULTS OF DIFFERENT EPOCHS ON TEMPERATURE AND WIND

Dataset	Temperature		Wind	
	MAE	RMSE	MAE	RMSE
Epoch : 60	0.921	1.532	1.268	1.985
Epoch : 70	0.909	1.513	1.252	1.962
Epoch : 80	0.891	1.488	1.239	1.947
Epoch : 90	0.868	1.456	1.223	1.928

TABLE III
THE RESULTS OF DIFFERENT EPOCHS ON CLOUD COVER AND HUMIDITY

Dataset	Cloud cover		Humidity	
	MAE	RMSE	MAE	RMSE
Epoch : 60	1.532	2.475	0.458	0.682
Epoch : 70	1.524	2.458	0.440	0.681
Epoch : 80	1.512	2.439	0.447	0.677
Epoch : 90	1.498	2.423	0.439	0.674

hidden units 32 and 64, similar performance are obtained. Therefore, our proposed DSTGNN is effective for different number of hidden units in meteorological prediction.

K. Runtime

We evaluate the runtime of one epoch on the four different datasets to compare DSTGNN against a baseline method, CLCRN. From Fig. 12, it can be seen that the proposed DSTGNN consistently has a lower runtime for one epoch compared to CLCRN on the various datasets. It is to be noted that the DSTGNN framework does not incorporate an encoder-decoder structure for local spatial-temporal information learning.

VI. CONCLUSION

In this paper, we analyze and discuss the limitations of existing literature which only utilizes local features and fails to adequately consider dynamic meteorological situation. We propose the DSTGNN framework which models global features and multiple local features to jointly learn the global-local features. For the global features, we design a random throwedge

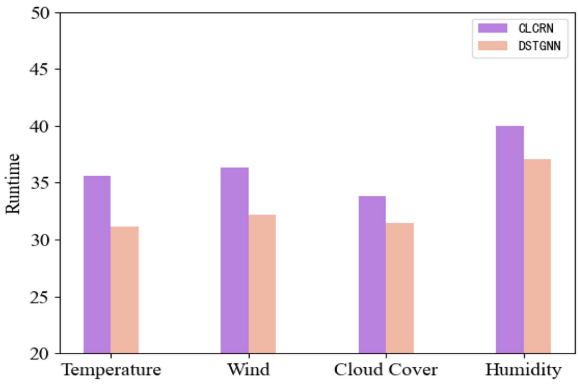


Fig. 12. Runtime comparison of one epoch.

module during the GNN neighborhood propagation process to extract the global features and adapt to the dynamic situation. We also establish convolution operation module to learn the features. Next, we perform information fusion on the two modules to capture sufficient features. Moreover, we use a graph ODE network and utilize coordinate information to obtain the long-term features and coordinate relationships. For the local features, we first construct a GNN for graph embedding. Then, we design another GNN to integrate into a GRU and jointly utilize the coordinate information to explore the features and coordinate relationships. Finally, we combine the global and local features using a global-local features learning layer for meteorological prediction. Extensive experimental results show that the proposed DSTGNN framework is superior the SOTA models.

REFERENCES

[1] R. B. Alley, K. A. Emanuel, and F. Zhang, "Advances in weather prediction,," *Science*, vol. 363, no. 6425, pp. 342–344, 2019.
[2] B. Klein, L. Wolf, and Y. Afek, "A dynamic convolutional layer for short range weather prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4840–4848.
[3] Z. Chang, X. Zhang, S. Wang, S. Ma, and W. Gao, "STAM: A SpatioTemporal attention based memory for video prediction," *IEEE Trans. Multimedia*, vol. 25, pp. 2354–2367, 2023.
[4] S. Mehrkanoon, "Deep shared representation learning for weather elements forecasting," *Knowl. Based Syst.*, vol. 179, pp. 120–128, 2019.
[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.
[6] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018.
[7] Y. Rong, W. Huang, T. Xu, and J. Huang, "DropEdge: Towards deep graph convolutional networks on node classification," in *Proc. Int. Conf. Learn. Representations*, 2020.
[8] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
[9] Y. Chen, Y. Hu, K. Li, C. K. Yeo, and K. Li, "Approximate personalized propagation for unsupervised embedding in heterogeneous graphs," *Inf. Sci.*, vol. 600, pp. 287–300, 2022.
[10] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 1234–1241.
[11] W. Long et al., "Unified spatial-temporal neighbor attention network for dynamic traffic prediction," *IEEE Trans. Veh. Technol*, vol. 72, no. 2, pp. 1515–1529, Feb. 2023.
[12] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Representations*, 2018.
[13] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 17804–17815.
[14] H. Lin, Z. Gao, Y. Xu, L. Wu, L. Li, and S. Z. Li, "Conditional local convolution for spatio-temporal meteorological forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 7470–7478.
[15] A. Grover, A. Kapoor, and E. Horvitz, "A deep hybrid model for weather forecasting," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 379–386.
[16] Z. Lin, M. Li, Z. Zheng, Y. Cheng, and C. Yuan, "Self-attention ConvLSTM for spatiotemporal prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11531–11538.
[17] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.
[18] Y. Chen, X. Zou, K. Li, K. Li, X. Yang, and C. Chen, "Multiple local 3D CNNs for region-based prediction in smart cities," *Inf. Sci.*, vol. 542, pp. 476–491, 2021.
[19] G. Terrén-Serrano and M. Martinez-Ramon, "Deep learning for intra-hour solar forecasting with fusion of features extracted from infrared sky images," *Inf. Fusion*, vol. 95, pp. 42–61, 2023.
[20] Y. Meng, F. Gao, E. Rigall, R. Dong, J. Dong, and Q. Du, "Physical knowledge-enhanced deep neural network for sea surface temperature prediction," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4203013.
[21] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.
[22] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 914–921.
[23] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, arXiv: *1709.04875*.
[24] J. Han, H. Liu, H. Xiong, and J. Yang, "Semi-supervised air quality forecasting via self-supervised hierarchical graph neural network," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 5230–5243, May 2023.
[25] L. Chen, W. Shao, M. Lv, W. Chen, Y. Zhang, and C. Yang, "AARGNN: An attentive attributed recurrent graph neural network for traffic flow prediction considering multiple dynamic factors," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17201–17211, Oct. 2022.
[26] Y. Chen, K. Li, C. K. Yeo, and K. Li, "Traffic forecasting with graph spatial–temporal position recurrent network," *Neural Netw.*, vol. 162, pp. 340–349, 2023.
[27] Z. Fang, Q. Long, G. Song, and K. Xie, "Spatial-temporal graph ODE networks for traffic flow forecasting," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2021, pp. 364–373.
[28] F. Li, H. Yan, G. Jin, Y. Liu, Y. Li, and D. Jin, "Automated spatio-temporal synchronous modeling with multiple graphs for traffic prediction," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, 2022, pp. 1084–1093.
[29] S. Rasp, P. D. Dueben, S. Scher, J. A. Weyn, S. Mouatadid, and N. Thuerey, "WeatherBench: A benchmark data set for data-driven weather forecasting," *J. Adv. Model. Earth Syst.*, vol. 12, no. 11, 2020, Art. no. e2020MS002203.
[30] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 922–929.
[31] B. Jing, H. Tong, and Y. Zhu, "Network of tensor time series," in *Proc. Web Conf.*, 2021, pp. 2425–2437.
[32] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Proc. 25th Int. Conf. Neural Inf. Process.*, Springer, 2018, pp. 362–373.

Yibi Chen received the PhD degree in computer science from Hunan University, China, in 2024. He is currently a postdoctoral researcher with Hunan University. His research interests include deep learning, data mining, graph representation learning, and spatial-temporal prediction.

Kenli Li (Senior Member, IEEE) received the PhD degree in computer science from the Huazhong University of Science and Technology, China, in 2003. He is currently the vice president with Hunan University and also is director with National Supercomputing Center in Changsha. He is a full professor of computer science and technology with Hunan University. His major research areas include parallel computing, deep learning, and distributed computing. He has published more than 300 research papers in international conferences and journals.

Chai Kiat Yeo received the BEng (1st class Hons.) and MSc degrees in electrical engineering from the National University of Singapore, and the PhD degree from the School of Electrical and Electronics Engineering, Nanyang Technological University (NTU), Singapore. She was an assistant principal engineer with Singapore Technologies Electronics and Engineering Limited prior to joining NTU in 1993. She is an associate professor and was the deputy director of Centre for Multimedia and Network Technology (CeMNet) and the associate chair (Academic) with the School of Computer Science and Engineering, NTU. She is currently the deputy director and programme lead of Singtel Cognitive and Artificial Intelligence Lab for Enterprises@NTU. Her current research interests include anomaly detection, machine learning, artificial intelligence, predictive operational analytics, ad hoc, and mobile networks.

Keqin Li (Fellow, IEEE) is a SUNY distinguished professor of computer science with the State University of New York. He is also a National distinguished professor with Hunan University, China. His current research interests include cloud computing, fog computing, and mobile edge computing. He has authored or coauthored more than 900 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He holds nearly 70 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the world's top five most influential scientists in parallel and distributed computing in terms of both single-year impact and career-long impact based on a composite indicator of Scopus citation database. He is a fellow of the AAAS and AAIA. He is also a member of Academia Europaea (Academician of the Academy of Europe).