

Prediction of three-dimensional ocean temperature in the South China Sea based on time series gridded data and a dynamic spatiotemporal graph neural network

Feng Nan¹, Zhuolin Li¹, Jie Yu¹, Suixiang Shi², Xinrong Wu², Lingyu Xu^{1*}

¹ Department of Computer Engineering and Science, Shanghai University, Shanghai 200444, China

² Key Laboratory of Digital Ocean, National Marine Data and Information Service, Tianjin 300171, China

Received 16 February 2023; accepted 3 May 2023

© Chinese Society for Oceanography and Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

Ocean temperature is an important physical variable in marine ecosystems, and ocean temperature prediction is an important research objective in ocean-related fields. Currently, one of the commonly used methods for ocean temperature prediction is based on data-driven, but research on this method is mostly limited to the sea surface, with few studies on the prediction of internal ocean temperature. Existing graph neural network-based methods usually use predefined graphs or learned static graphs, which cannot capture the dynamic associations among data. In this study, we propose a novel dynamic spatiotemporal graph neural network (DSTGN) to predict three-dimensional ocean temperature (3D-OT), which combines static graph learning and dynamic graph learning to automatically mine two unknown dependencies between sequences based on the original 3D-OT data without prior knowledge. Temporal and spatial dependencies in the time series were then captured using temporal and graph convolutions. We also integrated dynamic graph learning, static graph learning, graph convolution, and temporal convolution into an end-to-end framework for 3D-OT prediction using time-series grid data. In this study, we conducted prediction experiments using high-resolution 3D-OT from the Copernicus global ocean physical reanalysis, with data covering the vertical variation of temperature from the sea surface to 1 000 m below the sea surface. We compared five mainstream models that are commonly used for ocean temperature prediction, and the results showed that the method achieved the best prediction results at all prediction scales.

Key words: dynamic associations, three-dimensional ocean temperature prediction, graph neural network, time series gridded data

Citation: Nan Feng, Li Zhuolin, Yu Jie, Shi Suixiang, Wu Xinrong, Xu Lingyu. 2024. Prediction of three-dimensional ocean temperature in the South China Sea based on time series gridded data and a dynamic spatiotemporal graph neural network. *Acta Oceanologica Sinica*, 43(7): 26–39, doi: 10.1007/s13131-023-2252-0

1 Introduction

Ocean temperature is a physical quantity that reflects the thermal state of seawater (Sun et al., 2021) and is an important parameter in the energy balance system of the Earth's surface (Zhang et al., 2017). It plays an irreplaceable role in momentum, energy, and water exchange between the ocean and atmosphere (Sumner et al., 2003; Sun et al., 2019; Wentz et al., 2000). The seawater temperature field profoundly affects the distribution of global fishery resources, the direction of marine pollution, and the development of natural resources. Global climate anomalies are also closely related to changes in seawater temperature (Xiao et al., 2018). Therefore, the real-time forecasting of ocean temperatures is crucial in many fields (Solanki et al., 2015). However, owing to the influence of many uncertain factors, such as radiation, heat flux density, and offshore surface wind, the forecast accuracy is unsatisfactory (Zhang et al., 2020b).

Currently, ocean temperature prediction methods can be roughly classified into two categories (Patil et al., 2016): physics-based numerical prediction (Mendoza et al., 1997) and data-driven prediction methods. The former has shown considerable promise in many studies of seasonal sea surface temperature (SST) forecasting, with commonly used techniques such as hy-

brid coupled modelling (HCM) and linear inverse modeling (LIM). Of these, HCM assumes that detailed atmospheric structure is unimportant to seasonal SST forecasting, and therefore fully nonlinear ocean general circulation models (OGCMs) are coupled to simpler, often statistical, atmospheric models (Neelin, 1990). In addition to this, the development of coupled atmospheric circulation models (CGCMs) and initialization schemes significantly improves the prediction skill of CGCMs in seasonal SST forecasting (Barnston et al., 2019), e.g., predicting tropical SST, especially that of El Niño-Southern Oscillation, and several leading CGCMs can produce skillful ENSO prediction, with the anomaly correlation coefficients (ACCs) for ENSO-related SST indexes reaching 0.6 at the 12-month lead times (Luo et al., 2005). However, despite this, physics-based numerical model attempts to use a set of complex equations to describe ocean temperature variability patterns (Yang et al., 2018). Depending on the required initial and boundary conditions, prediction models are built using kinetics and thermodynamics, and the physical states are described using partial differential equations (Sun et al., 2021). For example, using a suite of 13 coupled atmosphere-ocean models to predict seasonal variations in sea surface temperature anomalies (Krishnamurti et al., 2006), the coupled mod-

Foundation item: The National Key R&D Program of China under contract No. 2021YFC3101603.

*Corresponding author, E-mail: xly@shu.edu.cn

el showed an improvement of about 10% in predictive accuracy compared to individual models, as assessed by comparing root mean square error (RMSE) and mean absolute error (MAE). For predicting the variability of tropical Atlantic sea surface temperature using the coupled ocean-atmosphere global climate model (GCM) (Stockdale et al., 2006), the relative error can be controlled at around 0.1. However, these equations are often complex and require a significant computational effort. Data-driven prediction methods fall into three main categories as follows.

(1) Traditional statistical learning methods. These methods include regression analysis (Laepfle and Jewson, 2007), empirical canonical correlation analysis (Collins et al., 2004), and Markov models (Xue and Leetmaa, 2000). Although these methods can predict the trend of ocean temperature to some extent, they are mainly applied to short- and medium-term predictions and have low overall prediction accuracy owing to the limitation of the prediction time length.

(2) Machine learning methods. Support vector machines (Lins et al., 2013) and genetic algorithms (Garcia-Goriz and Garcia-Sanchez, 2007), have been applied for SST prediction. Compared with traditional methods, these methods can prevent multiple minima problems associated with nonlinearity and tend to have better prediction accuracy. However, the performance of these machine-learning methods depends on a set of parameters in the associated learning problem. Choosing appropriate values for the model is a daunting task.

(3) Deep Learning methods. With the continuous innovation of deep learning techniques and algorithms, neural networks have been widely used for SST prediction owing to their flexibility and powerful modeling capabilities (Zhang et al., 2017). For example, artificial neural networks (ANN) have been used to predict SST in the northeastern Arabian Sea (Aparna et al., 2018) showed that the model error was $\leq 0.5^\circ\text{C}$ for more than 75% of the time. A nonlinear autoregressive (NAR) neural network was used to predict the monthly SST at six different locations over the Southern Indian Ocean (Patil et al., 2013), and the correlation coefficient between the predicted and actual values was above 0.9, with MSE and MAE values less than 0.23°C and 0.38°C , respectively. The gate recurrent unit encoder-decoder (GED) model generated (Xie et al., 2020) by combining gate recurrent unit

(GRU) and the attention mechanism achieved good results for multi-scale SST prediction, the MSE of GED for daily average SST forecast was about 2% lower than that of single-day forecast by fully connected long short-term memory (FC-LSTM), and the difference for a 7-d forecast increased to 6%. However, these methods only consider the temporal correlation of SST and ignore the spatial correlation of the data, which limits the prediction performance of SST to some extent. Therefore, capturing spatiotemporal correlations in data has received increasing attention. For example, the MHA model (Li et al., 2022) captures different levels of spatiotemporal correlations using a triple attention mechanism. The GSTGAT model uses a global graph attention network module to capture the global dynamic correlations of nodes (Gao et al., 2023).

Recently, graph neural networks (GNNs) have received extensive attention owing to their good spatial extraction capabilities with qualities such as spatial substitution invariance, local connectivity, and synthesis (Kipf and Welling, 2017). Current GNNs usually require a given predefined graph or the construction of a correlation graph between nodes by geographic location (Xiao et al., 2019), that is, the adjacency matrix is static. However, the spatial dependence between three-dimensional ocean temperature (3D-OT) data is affected by a variety of factors, resulting in a dynamic change in the spatial dependence between the nodes over time. Figure 2 demonstrates the changes in the two-year real seawater temperature values at three points (Fig. 1b) in our experimental region. As shown in Fig. 2a, the trends at Points A, B, and C show an overall positive correlation over time, that is, the curves rise or fall together most of the time. Next, we selected several typical moments to illustrate the dynamic changes of spatial dependence among the data: (1) as shown in Fig. 2b, Points A and B have high correlation, while Point C has a negative correlation with Point A; (2) in Fig. 2c, Points A, B, and C have positive correlations, but the positive correlation between Point A and Point B is stronger than that between Point A and Point C; (3) over time (Fig. 2d), the positive correlation between Point A and Point C is stronger than that between Point A and Point B. Thus, this study models spatial dependencies from the perspective of combining static and dynamic dependencies, which correspond to two types of graph matrices, that is, the static graph

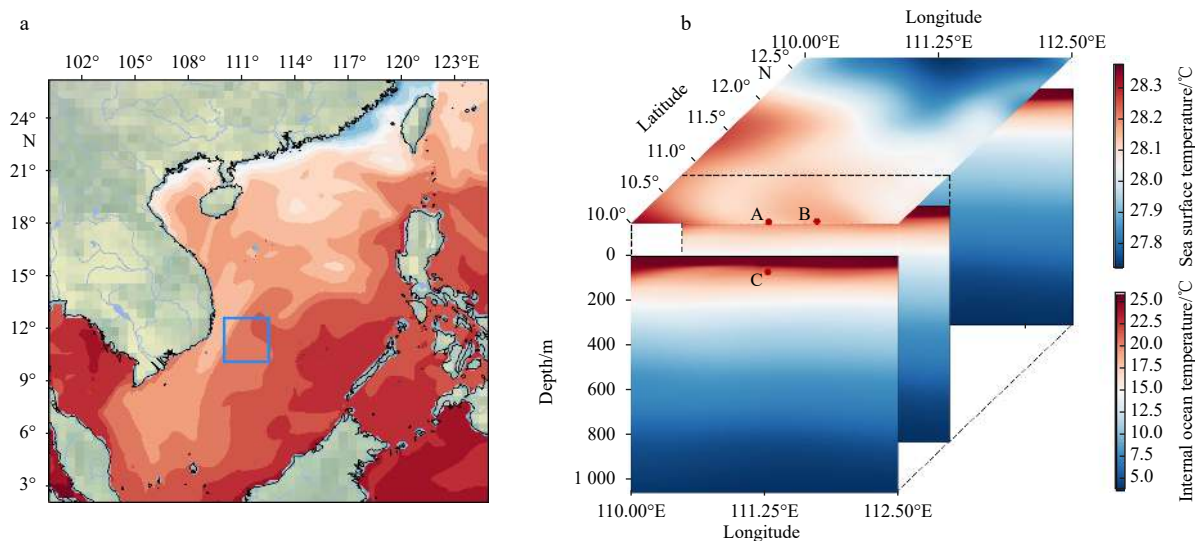


Fig. 1. Location of the experimental area (a, blue box) and a 3D representation of the experimental area, displaying average temperatures from 2000 to 2019 (b). In b, the color bar above indicates the sea surface temperature and the color bar below indicates the internal ocean temperature, where Points A, B, and C are data nodes, with Points A (10°N , 111.25°E) and B (10°N , 112°E) located at the sea surface, and Point C (10°N , 111.25°E) located 30 m below the sea surface.

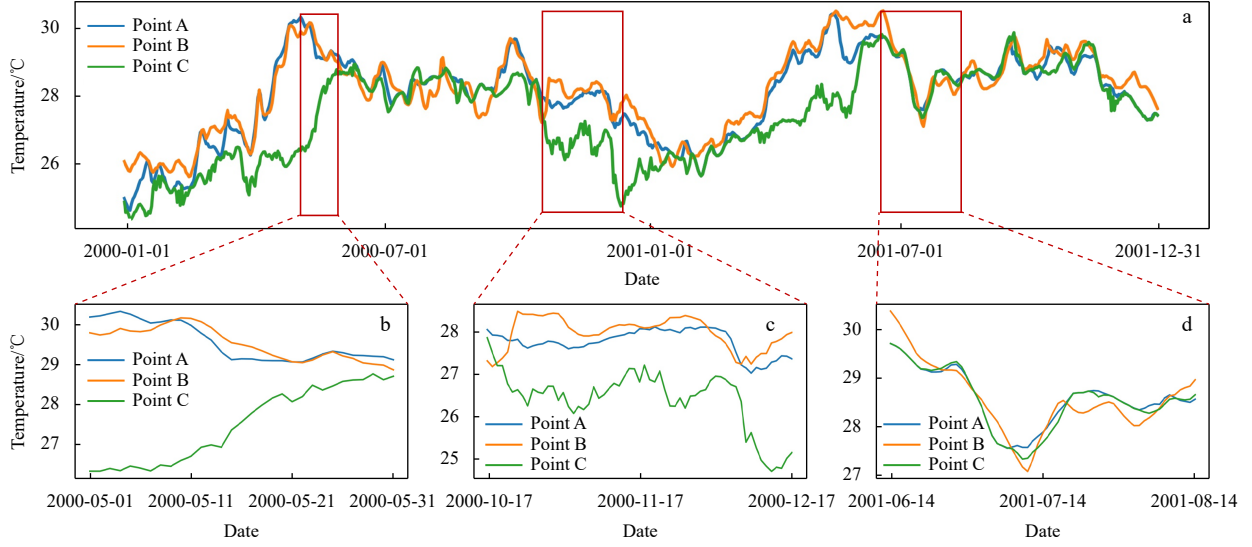


Fig. 2. Real seawater temperature profiles at three points in the experimental area. Points A (10°N , 111.25°E) and B (10°N , 112°E) are located at the sea surface, and Point C (10°N , 111.25°E) is located 30 m below the sea surface, as shown in Fig. 1. Date format: YYYY-MM-DD.

and dynamic graph matrices.

In addition, these studies only predict the SST, but the prediction of the ocean internal temperature is also important, as it covers approximately three-quarters of the Earth's surface. Many climatic phenomena, such as tsunamis, rainfall, and typhoons, are affected by ocean temperature. To achieve an accurate prediction of 3D-OT, a multi-layer convolutional LSTM network (Zhang et al., 2020a) was used to predict 3D-OT for the first time; however, the periodicities and fluctuations of ocean temperature at different depths have large differences, resulting in relatively poor model prediction accuracy. Meanwhile, the stereo spatial and temporal 4-D convolution model (Zuo et al., 2022) uses 3-D convolution to extract spatiotemporal features to improve accuracy.

To address the above problems, a dynamic spatiotemporal graph neural network (DSTGN) was proposed in this study for 3D-OT prediction. First, the static and dynamic graph learning layers of the model can learn static and dynamic graph matrices from the data, respectively. In places where the static graph was constant, it was used to capture the static dependencies of the data nodes. Dynamic graphs change and are used to capture dynamic dependencies. In addition, we designed a dynamic graph-learning method to mine the changing dependencies among the variables. Specifically, we fuse static node embeddings and dynamic inputs, based on which we design a multi-head adjacency mechanism to extract associations efficiently. The main contributions of this study are as follows.

(1) We designed a dynamic graph learning method that can capture the changing spatial dependencies between data nodes without any prior knowledge, and thus mine the evolution of associations from the data. This expands the application scenario of the GNN and plays an important role in obtaining unknown relationships.

(2) We designed an end-to-end graph learning framework that integrates dynamic graph learning methods with temporal convolution and graph convolution networks to model 3D-OT dependencies in both the temporal and spatial dimensions.

(3) The proposed method was tested on the South China Sea dataset, and the daily average 3D-OT data for the last 20 years were selected to form the 3D time-series gridded data. Com-

pared with the five other mainstream time-series methods, our method achieved the best results in all cases.

The remainder of this paper is organized as follows. Section 2.1 introduces the definition of the 3D-OT prediction problem. Section 2.2 outlines the overall framework and then introduces static graph learning, dynamic graph learning, temporal convolution, graph convolution, and output modules. Section 3 presents the experimental setup, performance evaluation, comparison, and discussion of the results. Section 4 discusses the maximum predictable time and the limitations of DSTGN. Section 5 summarizes the research presented in this paper.

2 Methods

2.1 Problem definition

Ocean temperature prediction refers to prediction of future temperatures based on historical temperature data. Among them, there are many ways to obtain ocean data. This paper uses the Copernicus Global Ocean Physical Reanalysis dataset. In this section, we describe the ocean temperature prediction problem. Given a series of fully observed 3-D time-series data $\mathcal{X} = \{\mathbf{X}_{:,0}, \mathbf{X}_{:,1}, \dots, \mathbf{X}_{:,t}, \dots\}$, where $\mathbf{X}_{:,t} = \{x_{1,t}, x_{2,t}, \dots, x_{i,t}, \dots, x_{N,t}\}^T \in \mathbb{R}^{N \times 1}$ are the observations of the N nodes at time step t . Our goal is to find a function F that predicts the data at the next τ steps of data based on the historical data of the past T steps:

$$\{\mathbf{X}_{:,t+1}, \mathbf{X}_{:,t+2}, \dots, \mathbf{X}_{:,t+\tau}\} = F(\mathbf{X}_{:,0}, \mathbf{X}_{:,1}, \dots, \mathbf{X}_{:,t-T+1}). \quad (1)$$

Multi-node ocean temperature prediction must consider the spatial dependence. Therefore, we used a graph-based approach to predict 3D-OT. The association graph of 3-D temporal data in this study is defined as $\mathbf{G} = (V, E)$, where V is the set of sensor nodes, E is the set of edges, and N is the number of nodes observing 3-D data. In addition, the adjacency matrix derived from the graph is denoted as $\mathbf{A} \in \mathbb{R}^{N \times N}$ with $A_{ij} > 0$ if $(v_i, v_j) \in E$ and $A_{ij} = 0$ if $(v_i, v_j) \notin E$.

2.2 DSTGN framework

Figure 3 depicts our proposed end-to-end 3D-OT prediction model framework, called the DSTGN. The model includes a stat-

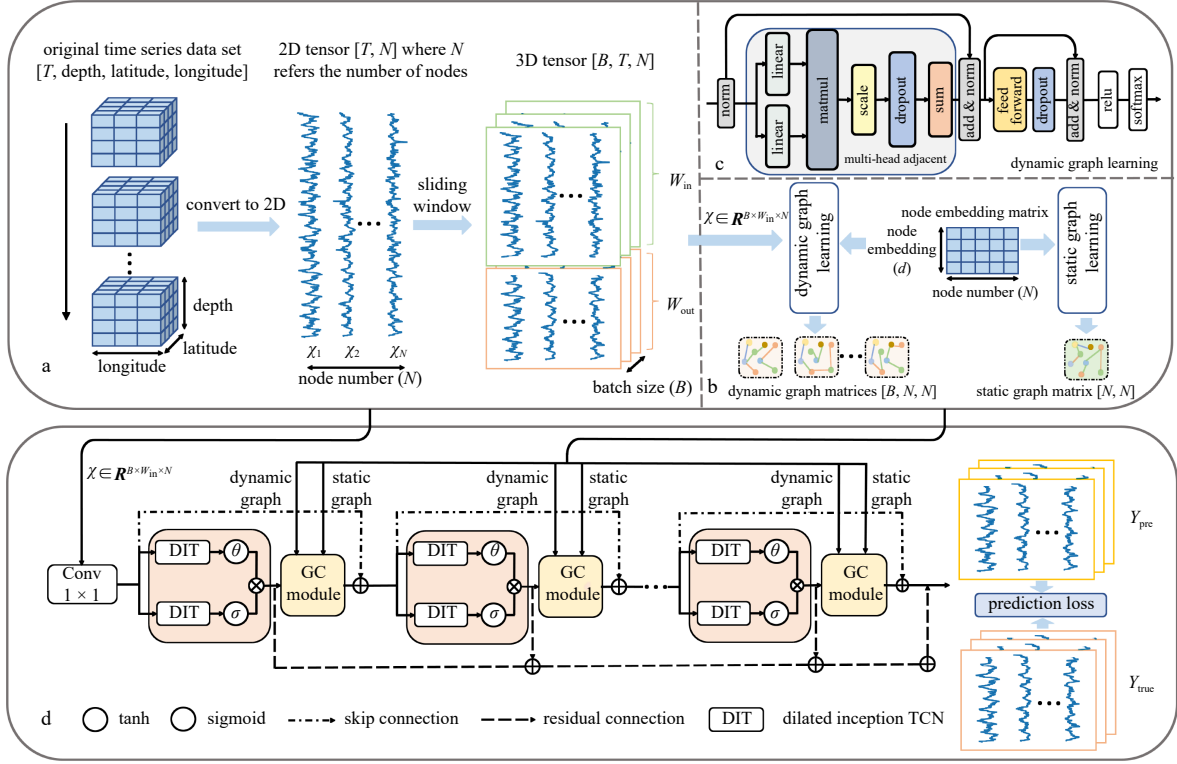


Fig. 3. The overall framework of DSTGN. a. The blue cube represents the raw input $\in \mathbb{R}^{T \times D \times L_1 \times L_2}$, which is first converted to a 2D tensor $\chi \in \mathbb{R}^{T \times N}$ and then sampled using a sliding window with input window size W_{in} and output window size W_{out} to obtain the input $\chi \in \mathbb{R}^{B \times W_{in} \times N}$ and the true label $Y_{true} \in \mathbb{R}^{B \times W_{out} \times N}$. b. The input data and node embedding matrix are dynamically learned to obtain the dynamic graph matrices, while the node embedding matrix is statically learned to obtain the static matrix. c. Details of the dynamic graph learning. d. The original data and the two types of graph adjacency matrices are transformed by K -layers of temporal convolution module and graph convolution module to extract features, and the features extracted by the temporal convolution module in each layer are linked to the output module through skip connections. Finally, the predicted result $Y_{pre} \in \mathbb{R}^{B \times W_{out} \times N}$ is obtained.

ic graph-learning layer, dynamic graph-learning layer, temporal convolution module, and graph convolution module. Two graph learning modules learn adaptively from the data to a static graph matrix and dynamic graph matrices, respectively to discover the two spatial dependencies hidden between the data nodes. A temporal convolution module was used to capture the temporal dependencies. The graph convolution module aggregates node information based on learned static and dynamic graphs to capture the spatial dependencies of the data. In addition, we add residual connections between the input of the temporal convolution module and the output of the graph convolution module to avoid the problem of gradient disappearance. Skip connections are added after each temporal convolution module. To obtain the output of the model, the final output module projects the hidden features onto the desired output dimension. The core components of our model are described in detail below.

2.2.1 Static graph learning layer

Most of the existing GCN-based SST prediction methods require a predefined adjacency matrix. Currently, the adjacency matrix given as a priori knowledge is mainly used to compute a predefined graph by similarity index evaluation (Sun et al., 2021) or a distance function (Zhang et al., 2022). This leads inevitably to the problem of high time and space complexity with $O(N^2)$. It means the computation and memory cost grows quadratically with the increase of graph size (Wu et al., 2020). Meanwhile, the predefined graph contains only the displayed spatial information, which is separated from the downstream tasks, and cannot contain the real spatial dependency information hidden between

the nodes, which can lead to a large degree of bias. To address these issues, we used node embedding to capture the static dependencies in the data (Wang et al., 2022). This approach does not require any prior knowledge and is learned end-to-end using stochastic gradient descent. As the loss function is backpropagated during model training, the adjacency matrix is adjusted to represent hidden associations between nodes.

Static graph learning: First, we randomly initialize the node embedding dictionary using a learnable parameter $E_s \in \mathbb{R}^{N \times d}$, where N denotes the number of sensors or nodes and d is the dimension of the node embeddings. We use Eq. (2) to calculate the dependencies between nodes, that is, adjacency matrix \hat{A} . It is noteworthy that we used the ReLU activation function to eliminate weak connections. The Softmax function was applied to normalize the self-adaptive adjacency matrix.

$$\hat{A} = \text{SoftMax}(\text{ReLU}(E_s \cdot E_s^T)). \quad (2)$$

2.2.2 Dynamic graph learning layer

As shown in Fig. 2, the spatial information of 3D-OT data is complex, the dependence between nodes is affected by various factors, such as offshore surface wind and radiation, and the dependence between data shows a complex nonlinear variation. In 3D-OT multi-node time-series data, there are not only long-term patterns between nodes but also short-term, dynamically changing patterns. Some works have considered capturing dynamic spatial dependencies (Guo et al., 2019; Shi et al., 2019). However, assuming dynamic spatial dependencies makes the model ex-

tremely hard to converge when we need to learn the graph structure at the same time. In our work, we designed a dynamic graph learning layer by first transforming \mathcal{X} into the same dimension \mathcal{X}_T as E_s using a linear layer, and then fusing these two components using Eq. (3). The proposed dynamic graph-learning module is then used to generate dynamic matrices based on the fusion results.

$$h_t = \text{LN}(W_s \cdot \mathcal{X}) + E_s, \quad (3)$$

where $\mathcal{X}_T = \text{LN}(W_s \cdot \mathcal{X})$, LN denotes LayerNorm, W_s is the weight matrix to be learned, the node embedding E_s as long-term information, and the node-level input \mathcal{X} as short-term information. Finally, we obtain the fusion results $h_T \in \mathbf{R}^{N \times d}$ with the time window T .

Dynamic graph learning: As shown in Fig. 3c, an attention mechanism (Vaswani et al., 2017) was introduced in this study with the goal of constructing association matrices between nodes. To better train the model (Xiong et al., 2020), we used a multi-head approach to construct the matrix and added skip connections and MLPs (Dong et al., 2021), which prevents overfitting and captures the dependencies between 3D-OT nodes more fully. As shown in Fig. 3c, we used the learnable parameters W_q and W_k to project h_t to the d_k dimension to obtain the Q and K matrices, respectively, where $Q, K \in \mathbf{R}^{N \times \text{head} \times d_k}$. Then, we performed parallel operations on each head in Q and K , using a scaled dot-product to compute the correlation between v_i and v_j , and then summed the multi-head matrices, as shown in Eq. (4) and Eq. (5):

$$\text{Adj}_{v_i, v_j}^{\text{head}_i} = \text{dropout} \left(\frac{Q_{\text{head}_i} K_{\text{head}_i}^T}{\sqrt{d_k}} \right), \quad (4)$$

$$O_T = \text{LN} \left(\sum_{i=1}^n \text{head}_i + G_r \cdot G_r^T \right). \quad (5)$$

In Eq. (4), we added dropout to improve the generalization ability of the model, where head_i is obtained by a linear layer projection: $\text{head}_i = W_h \text{Adj}_{v_i, v_j}^{\text{head}_i}$. $G_r = W_r h_T$ was used as the residual connection in Eq. (5) to prevent model degradation (He et al., 2016). Then, we used MLPs for feature projection, as shown in Eq. (6):

$$\tilde{Q}_T = \max(0, O_T W_1 + b_1) W_2 + b_2, \quad (6)$$

where b_2 denotes the bias vector. Finally, we used O_T as the residual connection, ReLU as the activation function (Glorot et al., 2011), and SoftMax for normalization to obtain the dynamic graph matrices $A_d^T \in \mathbf{R}^{N \times N}$, as shown in Eq. (7):

$$A_d^T = \text{SoftMax}(\text{ReLU}(\text{LN}(\tilde{Q}_T + O_T))). \quad (7)$$

2.2.3 Temporal convolution module

As shown in Fig. 3d, the temporal convolution module consists of two parallel dilated initial TCN (DIT) and gating units. In DIT, we introduce the inception mechanism and dilated causal convolutions. When the inception layers use multiple scales of 1D convolutional filters to capture historical information about the data, the dilated causal convolutions reduce the model complexity, and gating units are used to control the amount of information passed to the next module.

Dilated causal convolutions: First, choosing the right filter size is crucial for convolutional networks. Therefore, we introduced an inception mechanism (Szegedy et al., 2015). However, in contrast to the 2D convolutional kernels of several sizes used in that study, multiple filters with different sizes were adopted to extract temporal patterns with various ranges, considering the

different periodicity and volatility of different depth layers. Second, although standard convolutional networks have the features of gradient stability, parallel computation, and low memory requirements compared to RNNs, when processing long-term time-series data, we need to introduce a very deep network or a very large filter, which is not a good choice. To prevent information leakage, we have to ensure that the predictions of the current time step involve only historical information. We consider the introduction of dilated causal convolution (Yu and Koltun, 2016). We let the dilated factor of each layer grow exponentially at a rate of d . Assuming that the dilated factor of the initial layer is 1, the receptive field size of the n -layer dilated convolution with kernel size c is as follows:

$$R = \begin{cases} 1 + n(c-1), & \text{if } d = 1, \\ 1 + \frac{(c-1)(d^n - 1)}{d-1}, & \text{if } d > 1. \end{cases} \quad (8)$$

As seen in Eq. (8), the receptive field size of the network increases exponentially with the number of hidden layers at a rate of d . Therefore, the introduction of dilated convolution can reduce the complexity of the model compared to standard convolution approaches. In addition, based on the above work, we provide a 1D sequence of inputs $p \in \mathbf{R}^T$ and a set of filters; the inception layers of the dilated casual convolution are in the form shown in Eq. (9):

$$p = \text{concat}(p \times f_{1 \times k_1}, p \times f_{1 \times k_2}, \dots, p \times f_{1 \times k_\psi}), \quad (9)$$

$$p * f_{1 \times k}(t) = \sum_{s=0}^{k-1} f_{1 \times k}(s) p(t-d \times s), \quad (10)$$

where $*$ denotes the dilated causal convolution operation. The outputs of all filters are truncated to the same length according to the largest filter and concatenated across the channel dimension. The dilated convolution denoted by $p * f_{1 \times k}$ is defined by Eq. (10).

Gated TCN: To control the amount of information passed from the temporal convolution module to the next module, we followed two inception layers with Tanh and Sigmoid activation functions. As shown in Eq. (10), for a given input $\mathcal{X} \in \mathbf{R}^{N \times D \times S}$, we have the following form:

$$l = \theta(\Theta_1 * \mathcal{X} + b) \odot \sigma(\Theta_2 * \mathcal{X} + c), \quad (11)$$

where Θ_1 , Θ_2 , b , and c are the model parameters, \odot is the element-wise product, $\theta(\cdot)$ is the activation function of the outputs, and $\sigma(\cdot)$ is the ratio of information determined to be passed to the next layer.

2.2.4 Graph convolution module

The graph convolution module fuses the information of a node with information from its neighbors to obtain new node features. A diffusion convolution layer was proposed for traffic forecasting (Li et al., 2017), which has been shown to effectively capture the spatiotemporal information of the data and to model the diffusion process of the graph signal with K finite steps. The diffusion convolution layer is defined as follows:

$$U = \sum_{k=0}^K P^k X W_k, \quad (12)$$

where $X \in \mathbf{R}^{N \times D}$ denotes the input signals, $W_k \in \mathbf{R}^{D \times M}$ denotes the model parameters, $U \in \mathbf{R}^{N \times M}$ denotes the output, and $P^k \in \mathbf{R}^{N \times N}$ denotes the transformation matrix power. In the case of a graph, $P = A / \text{rowsum}(A)$, where $A \in \mathbf{R}^{N \times N}$ is the adjacency matrix.

In addition, inspired by the work of Liu et al. (2020), we de-

couple the information propagation and representation transformation operations to alleviate the transition smoothing problem in deeper graph neural networks. We use W_k as the information selection layer with the following equation:

$$U = W_k \text{concat}(P^1 X, \dots, P^k X, X), \quad (13)$$

where W_k is implemented with a 1×1 convolution, with input channel $r(s+1)$ and output channel r . In extreme cases, that is, when there are no dependencies among variables, aggregating information only adds useless noise to each node; however, Eq. (13) still preserves the information of the nodes by adjusting W_k to 0 for $P^1 X, \dots, P^k X$. To model the short- and long-term interaction patterns of 3D-OT, we replaced A with the two graph matrices \hat{A} and A_d^T learned from the static and dynamic graph modules.

2.2.5 Skip connection and output module

As shown in Fig. 3d, in the stacked graph and temporal convolution operations, each layer was connected to the output module through a skip connection. The skip connection is essentially a $1 \times h_i$ standard convolution, where h_i is the sequence length of the i -th input layer. The output module of the model is composed of two 1×1 standard convolutional layers that convert the number of input channels into the output dimension. For example, when predicting the ocean temperature for T consecutive days, the output dimension of the last layer in the output module is T .

The entire algorithmic process of the DSTGN is presented in Algorithm 1. As we can see, our model does not require an a priori graph structure and only needs to initialize the node embedding. Our model constructs static and dynamic graph matrices based on node embeddings and input data, respectively. The input data and graph matrix were subjected to feature extraction and aggregation using temporal convolution and graph convolution modules. In the stacked convolution operation, each layer is connected to the output module via skip connections. Finally, we update the model parameters and node embeddings using back-propagation based on the predicted loss L_{loss} .

Algorithm 1. Framework for DSTGN

Input: Dataset $O \in R^{N \times \text{Len}}$
Parameter: learning rate γ ; batch size b ; iteration steps iter; module layers k ; the initialized DSTGN model with θ and word embedding with M_s .
Output: prediction value \hat{Y} .

- 1: **repeat**
- 2: **For** i in 1: iter **do**
- 3: Sample a batch ($X \in R^{B \times N \times H}$, $Y \in R^{B \times N \times L}$) from O ;
- 4: Compute received field using Eq. (7);
- 5: Compute $\hat{A} = \{M_s\}$ using Eq. (2);
- 6: Compute $A_d^T = \{M_s, \mathcal{X}\}$ using Eqs (3)-(6);
- 7: **for** j in 1: k **do**
- 8: Compute $H^j = \{X\}$ using Eq. (8);
- 9: Compute $Z_j^T = \{H^j, \hat{A}, A_d^T\}$ using Eq. (12);
- 10: Compute $\mathcal{X} = Z_j^T$ and $\hat{Y}_+ = H^j$;
- 11: **end for**
- 12: Compute $L = \text{loss}(\hat{Y}, Y)$;
- 13: Update model parameters θ and M_s according to L and the learning rate γ ;
- 14: **end for**
- 15: $i = i + 1$;
- 16: **until convergence**

3 Experiment

3.1 Dataset

We used high-resolution 3D-OT data from the Copernicus Global Ocean Ensemble Physical Reanalysis, which was obtained from the website <https://resources.marine.copernicus.eu/products>, covering the daily and monthly average seawater temperature of the global ocean from January 1, 1993, to December 31, 2019. Depths range from 0.5 m underwater to 5 902.1 m, with 75 vertical depths, covering a global ocean range of 98.875°S–98.875°N, 0.125°E–180°–0.125°W. We selected the daily average temperature data (7 305 days) from January 1, 2000 to December 31, 2019 for the local area of the South China Sea. As shown in Fig. 1, the selected data area was 10°N–12.5°N, 110°E–112.5°E, with a resolution of 0.25°, and the depths are 0.51 m, 30.87 m, 97.04 m, 508.64 m, 1 045.85 m (for the convenience of writing, we round the depths to 0 m, 30 m, 100 m, 500 m, 1 000 m). Excluding some grid points in the test area lacking data, there were 117 data points in each layer, and a total of 585 points; the temperature of each point corresponds to a unique longitude, dimension, and depth, which is a 3-D gridded data point. The sea temperature changes at different depths are shown in Fig. 4. As the depth increases, the overall temperature of sea water decreases, and the periodicity of seawater temperature becomes increasingly less obvious.

3.2 Baseline

In this section, we comprehensively evaluate the performance of the proposed DSTGN model and five baseline methods on the selected datasets. The five baseline methods used for comparison are as follows.

(1) Support vector regression (SVR) is widely used for time series forecasting.

(2) FC-LSTM consists of an LSTM layer and fully connected layer. The LSTM layer models time-series relationships. The fully connected layer maps the output of the LSTM layer to the final forecast.

(3) Temporal pattern attention long short-term memory (TAP-LSTM) is a recursive attention network that uses a novel attention mechanism to select relevant time series, allowing the model to learn the interdependencies between multiple SST variables over all time series.

(4) Dual self-attention network (DSANet) utilizes two parallel convolutional components to capture a complex mixture of global and local temporal patterns, and employs a self-attentive module to model the dependencies between multiple sequences.

(5) Graph WaveNet is a graph neural network approach that uses a new adaptive dependency matrix and learns through node embedding, allowing the model to accurately capture hidden spatial dependencies in SST catch data.

3.3 Experiment settings and metrics

The experiments were conducted using an Intel (R) Core (TM) i5-11400F @2.60HZ, NVIDIA GeForce GTX 1660S, 16G RAM Windows 10 64-b operating system, and Python 3.6 environment. The deep development tool PyTorch was used. Specifically, the model parameters were optimized by choosing the Adam optimizer, the learning rate was initialized to 0.001, the learning rate was automatically adjusted using learning rate decay, the batch size of the input data was 32, and the number of iterations was set to 1 000. We first used $\chi_{\text{norm}} = \frac{\chi_{i,t}}{\max(\chi_{i,t})}$ to preprocess the data, where χ_{norm} is the normalized data, $\chi_{i,t}$ denotes the value of the

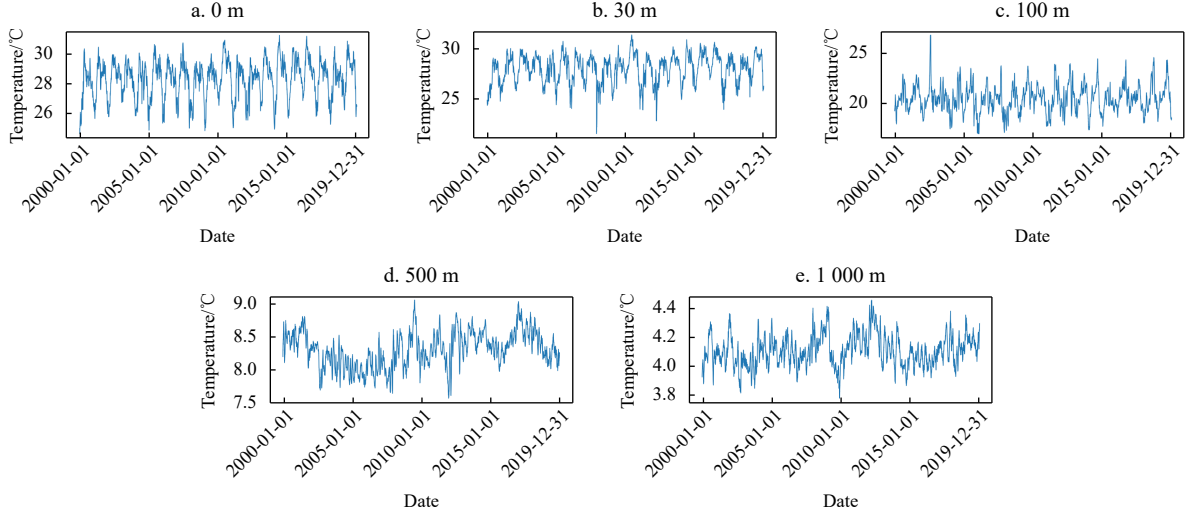


Fig. 4. Seawater temperature trends at different depths at the same location (10°N, 111.25°E). Date format: YYYY-MM-DD.

i -th point at time t , and $\max(\chi_{i,:})$ denotes the historical maximum value of the i -th point. Then, the data were divided into training, validation, and test sets at a ratio of 6:2:2 according to the time order. The data from January 2000 to December 2011 (4 353 d in total) are used as the training set, the data from January 2012 to December 2015 (1 431 d in total) are the validation set, and data from January 2016 to December 2019 (1 431 d in total) are the test set. We used a sliding window with a window size of 30 to sample the data and predict the future ocean temperature values at each point for 1, 3, and 7 consecutive days. We chose the mean absolute error (MAE), root mean square error (RMSE) and empirical correlation coefficient (CORR) as the evaluation criteria for the prediction model performance, which are defined as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\bar{y}_i - y_i|, \quad (14)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\bar{y}_i - y_i)^2}, \quad (15)$$

$$\text{CORR} = \frac{1}{n} \sum_{i=1}^n \frac{\sum_t (y_{it} - \text{mean}(y_i)) (\bar{y}_{it} - \text{mean}(\bar{y}_i))}{\sqrt{\sum_t (y_{it} - \text{mean}(y_i))^2 (\bar{y}_{it} - \text{mean}(\bar{y}_i))^2}}, \quad (16)$$

where y_i and \bar{y}_i are the true and predicted values, respectively, of the i -th point. MAE is the average of the absolute error between the true and predicted values, which reflects the overall prediction performance. CORR is used to measure the degree of linear correlation between variables and takes a value between 0 and 1. Compared to MAE, RMSE is more sensitive to values with larger prediction errors in the data. For both MAE and RMSE, a smaller value indicated a better model effect and a higher value for CORR is better

3.4 Comprehensive comparison of the experimental results

We conducted experiments using a dataset at multiple prediction scales to verify the validity of the proposed model. Table 1 shows the overall prediction performance, that is, the average

Table 1. Comparison of the prediction results of different model

Model	Metrics	Metrics value		
		Scale = 1 d	Scale = 3 d	Scale = 7 d
SVR	MAE	0.091 5	0.128 9	0.202 2
	RMSE	0.256 0	0.297 6	0.396 2
	CORR	0.982 7	0.978 9	0.922 2
FC-LSTM	MAE	0.183 1	0.198 0	0.247 2
	RMSE	0.282 4	0.304 6	0.388 6
	CORR	0.883 6	0.912 5	0.865 3
TAP-LSTM	MAE	0.174 1	0.243 2	0.281 5
	RMSE	0.272 3	0.373 0	0.434 0
	CORR	0.907 5	0.835 0	0.818 1
DSANet	MAE	0.090 0	0.153 9	0.197 7
	RMSE	0.152 2	0.251 5	0.330 7
	CORR	0.990 3	0.974 1	0.946 8
Graph-WaveNet	MAE	0.071 8	0.109 7	0.191 8
	RMSE	0.116 0	0.180 1	0.310 4
	CORR	0.990 8	0.979 1	0.949 5
DSTGN	MAE	0.065 7	0.097 8	0.167 2
	RMSE	0.111 6	0.168 8	0.281 1
	CORR	0.991 2	0.978 9	0.951 2

Note: Bold numbers denote the best predictive results.

MAE and RMSE results for multiscale ocean temperature predictions in the South China Sea. The black font in the table indicates the best prediction and the underline indicates the suboptimal prediction. Several observations are worth highlighting based on these results.

As shown in Table 1, (1) DSANet that uses an attention mechanism to establish spatial correlations, is better than SVR and FC-LSTM because the latter does not model spatial information, which shows that proper spatial modeling is effective. (2) The two best-performing models are DSTGN and Graph WaveNet, both of which use graph neural networks to capture the spatial dependencies of 3D-OT nodes, thereby demonstrating the power of the graph structure in modeling spatial association. (3) Our DSTGN achieved the best performance at all prediction scales because our approach adaptively learns more realistic dependencies from the data using a combination of static graphs and dynamic graphs to capture static and dynamic dependencies from the data, respectively, thereby making full use of the spatiotemporal information.

Compared with the best baseline method, Graph WaveNet, our method reduces the MAE by 8.5%, 10.85%, and 12.83%, and RMSE by 3.79%, 6.27%, and 9.44% at prediction scales of 1 d, 3 d, and 7 d, respectively. This indicates that the performance of the DSTGN is improved more significantly compared to the other models as the prediction scale increases. In addition, as shown in Fig. 5, in terms of the error of the 30-d prediction for each of the next seven consecutive days, our model achieved the best results during each prediction day. This demonstrates the effectiveness of the DSTGN in combining static and dynamic dependencies.

3.5 Study on different ocean depths

To further verify the powerful ability of our proposed model to capture dynamic data changes, we stratified the experimental data by depth constructed a DSTGN model to simultaneously predict all data, while ensuring that other comparative models maintain the same operations as DSTGN. Finally, we analyzed the performance of the model at each depth layer. In terms of generating samples, we sampled the data in steps of one using a sliding window of input window size 30 to predict 3D-OT values for each stratum for the next seven consecutive days. In other words, we used the past 30 consecutive days of 3D-OT to forecast

the future 7 d of 3D-OT. We also used the MAE, RMSE and CORR metrics to evaluate the model performance. The results of the metric evaluation for the different models are shown in Table 2.

Based on Table 2 and Fig. 6, we can see that (1) the prediction errors of all models are much smaller than those of the other three ocean temperature layers at depths of 500 m and 1 000 m. This is because the internal ocean temperature was not affected by the time dimension of the surface layer. (2) Our DSTGN achieves the best prediction results for all ocean temperature layers, and the MAE of DSTGN is reduced by 2.7%, 5.63%, 4.6%, 10.26%, and 10.81% for the five 3D-OT layers, respectively, compared with the suboptimal model Graph WaveNet. The performance improvement of our model is more obvious with increasing depth.

Next, we show the fit of DSTGN on the test data and plot the true and predicted values on the same graph to visualize the performance of the predicted fit. As shown in Fig. 7, the model fits are given for three prediction scales of 0 m, 30 m, 100 m, 500 m, and 1 000 m depths at the 10°N and 111.25°E points. The variation in seawater temperature is influenced by several factors, and the trend and magnitude of seawater variation varies at dif-

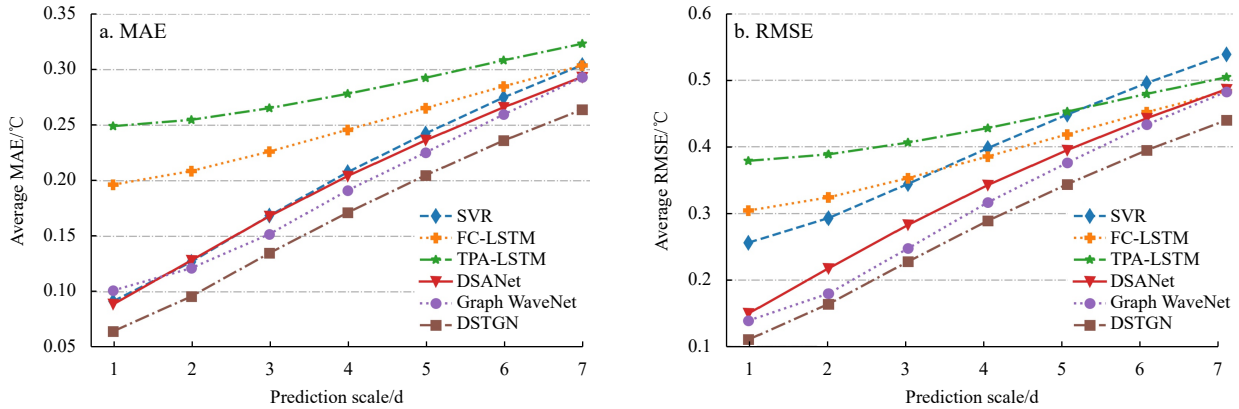


Fig. 5. Comparison of the prediction results of different models.

Table 2. Comparison of the prediction results of different models at different depths

Model	Metrics	Metrics value				
		Depth = 0 m	Depth = 30 m	Depth = 100 m	Depth = 500 m	Depth = 1 000 m
SVR	MAE	0.252 1	0.330 9	0.335 5	0.060 5	0.031 9
	RMSE	0.334 4	0.566 6	0.581 7	0.088 0	0.040 7
	CORR	0.947 8	0.938 2	0.940 4	0.923 7	0.834 6
FC-LSTM	MAE	0.272 9	0.324 4	0.288 1	0.057 0	0.025 1
	RMSE	0.361 8	0.466 5	0.389 9	0.076 1	0.032 6
	CORR	0.933 5	0.942 2	0.959 6	0.961 0	0.924 9
TAP-LSTM	MAE	0.278 2	0.356 0	0.349 8	0.064 5	0.027 2
	RMSE	0.371 0	0.516 8	0.483 7	0.086 6	0.035 7
	CORR	0.944 4	0.920 6	0.934 7	0.901 5	0.891 2
DSANet	MAE	0.282 2	0.331 5	0.298 3	0.057 1	0.026 0
	RMSE	0.372 0	0.489 0	0.411 4	0.078 3	0.034 1
	CORR	0.944 5	0.940 2	0.956 8	0.961 4	0.921 9
Graph WaveNet	MAE	0.240 7	0.287 7	0.258 5	0.046 8	0.022 2
	RMSE	0.321 4	0.429 2	0.364 4	0.066 0	0.030 4
	CORR	0.958 6	0.952 0	0.966 0	0.961 9	0.932 9
DSTGN	MAE	0.234 2	0.271 5	0.246 6	0.042 0	0.019 8
	RMSE	0.310 4	0.400 9	0.344 5	0.058 8	0.026 9
	CORR	0.960 4	0.956 1	0.969 6	0.964 5	0.943 8

Note: Bold numbers denote the best predictive results.

ferent depth layers. The seawater temperature trends at 0 m, 30 m, and 100 m below the sea surface have obvious periodicity, there are obvious drastic change amplitudes at specific times, and the fitting of change amplitudes is one of the prediction difficulties, which includes maximum amplitude, time lag offset, etc. The trend of sea temperature change at 500 m and 1 000 m below the sea surface has no obvious pattern, and the amplitude of the data change is relatively small, but the change is more frequent and drastic. DSTGN has a good fitting effect on all prediction scales of the five depth layers, which indicates that DSTGN has good adaptation to both periodic changes and extreme change trends and also shows the generalization of DSTGN to fit trends with different characteristics.

To provide a more intuitive understanding of the prediction performance of DSTGN, we display the temperature distribu-

tions of each depth layer for 4 d in the test set, along with the predicted temperature distributions and the absolute error distributions of DSTGN, as shown in Fig. 8. It is worth noting that the predicted results use the first day of the 7-d forecast from a 30-d input sequence, so that the prediction performance is not affected by time lag shift due to the accumulation of errors over time. In addition, the time corresponding to Figs 8a, b, c, and d are “Spring”, “Summer”, “Autumn”, and “Winter”, respectively. Visualizing the prediction performance in different seasons can help us better analyze the performance of DSTGN.

From Fig. 8, it can be observed that: (1) For the same depth, the sea temperature distribution has undergone significant changes in different seasons, but the temperature range has only undergone small changes, almost unaffected by the season. Considering that the experimental area is near the equator, where the

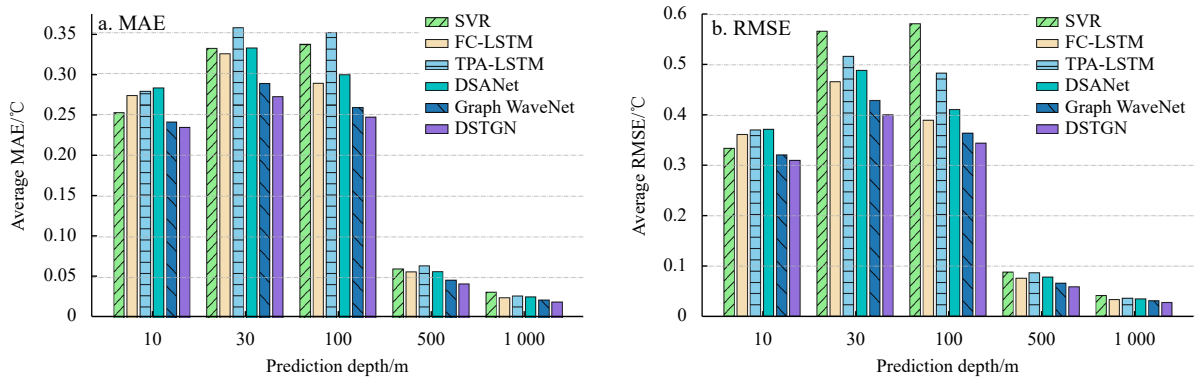


Fig. 6. Comparison of the prediction results of different models at different depths.

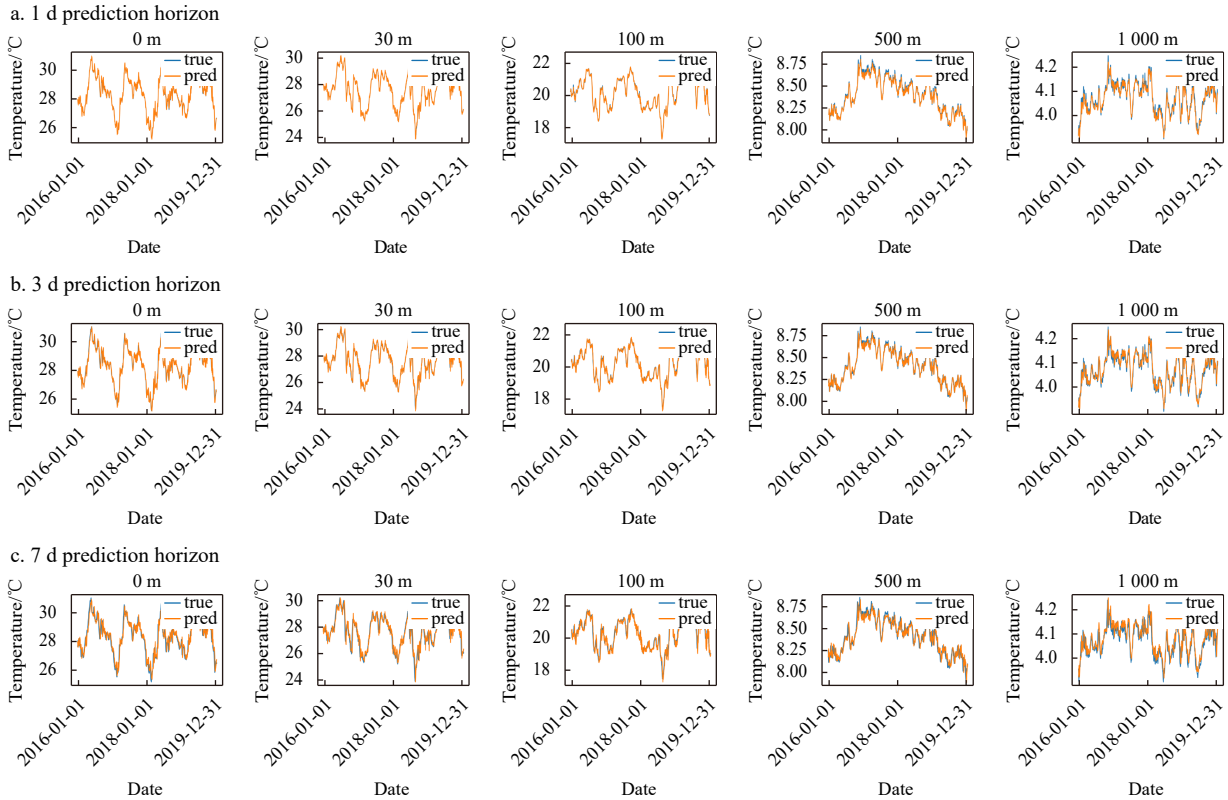


Fig. 7. Fit plot of average temperature predictions (pred) at different depths. The blue curve represents the average actual temperature in the experimental area, and the orange curve represents the average predicted temperature in the corresponding area. Date format: YYYY-MM-DD.

sun’s angle of incidence is large, the solar radiation is strong, and the stable climatic environment results in small differences in seawater temperature under different seasons, which is consistent with the reality. Meanwhile, the sea temperature distribution and temperature range at different depths are significantly different under the same season, with one of the factors being depth, i.e., the deeper the depth, the less the seawater is affected by surface heat conduction and convection. In addition, the movement and mixing of deep-sea water are mainly caused by ocean circulation and the rotation of the Earth, which requires a long time, so the temperature change of deep-sea water is much smaller than that of surface seawater. (2) For all seasons and all depths,

the predicted results of DSTGN achieved high accuracy, with absolute error less than 0.7°C at depths of 0 m, 30 m, and 100 m, and less than 0.2°C at depths of 500 m and 1 000 m. This indicates that DSTGN has high accuracy and stability and can effectively predict the changes of ocean temperature at different depths.

We also visualized the error distribution, as shown in Fig. 9, which shows the distribution of the MAE of models such as DSTGN over different ocean temperature layers, where darker colors represent smaller errors, brighter colors indicate larger errors, and the areas circled in red indicate that our model is significantly better than the others. It can be observed that SVR performs poorly in almost all depth layers except at 0 m, where it

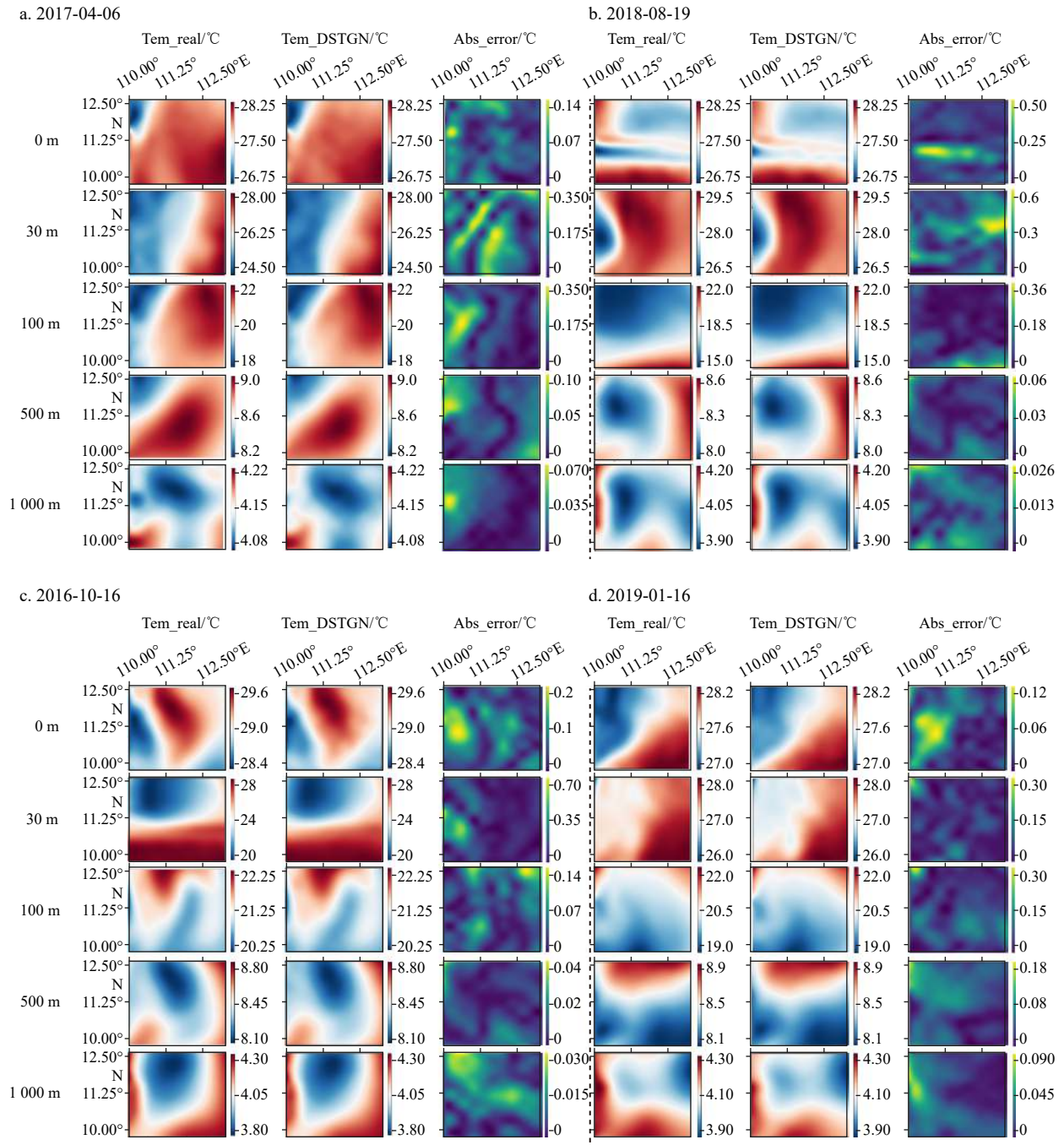


Fig. 8. Visualization of the 3D-OT distribution and the predicted results of DSTGN for different depth layers. “Tem_real” represents the true temperature, “Tem_DSTGN” represents the predicted temperature by DSTGN, and “Abs_error” represents the absolute error, which is defined as $|\text{Tem_DSTGN} - \text{Tem_real}|$.

shows a relatively small error. This is because SVR is essentially a linear model, and its modeling capability for non-linear relationships is weak. TPA-LSTM also performs poorly, and it is speculated that this is because TPA-LSTM only uses a single-layer LSTM as its basic component, which cannot capture the non-linear features and temporal dynamics in the 3D-OT data very well. FC-LSTM and DSANet have similar performance. FC-LSTM uses fully connected layers to process input sequences and can capture long-term dependencies and non-linear relationships, but it lacks spatial information. DSANet's two self-attention mechanisms can process global and local information separately, but it has not completely solved the problem of long-term dependencies. DSTGN and Graph WaveNet are the two best-performing models, and the difference lies in DSTGN's dynamic modeling of spatial dependencies. The errors of our model are smaller than those of other models for all ocean temperature layers, which further illustrates the effectiveness of our proposed model in spatiotemporal modeling.

4 Discussion

4.1 Maximum forecast time

In this section, we discuss the maximum predictable time of DSTGN. In fact, there is no unified standard for maximum predictable time. In our 3D-OT prediction task, we first fix the input window at 30 d and then observe the prediction performance by gradually increasing the prediction time up to 360 d (denoted as 30–360 d), while comparing it with the best baseline model Graph WaveNet to examine the effect of prediction time. Meanwhile, we fix the prediction time and observe the effect of input window size by increasing the input window size, in order to explore the maximum predictable time.

From Figs 10 and 11, we can observe the following: (1) In Fig. 10a, we set the input window to 30 d, and with the increase of prediction time, the error increases sharply at first and then stabilizes. However, DSTGN stabilizes earlier and overall has smaller prediction errors, with MAE staying below 0.7°C for a long period of time, as also shown in Fig. 11. (2) In Figs 10b and c, the input windows are set to 60 d and 90 d, respectively (denoted as 60–360 d and 90–360 d). It can be observed that with the same prediction time, larger input window leads to smaller prediction errors, because more historical information is provided as input, allowing the model to learn longer dependencies and stronger patterns, resulting in higher prediction accuracy. (3) We show the boxplots of prediction errors for DSTGN and Graph WaveNet in Fig. 10 under the three input window settings, as depicted in Fig. 11. It can be seen that with the same input window, the larger the prediction time, the larger the overall error, as errors accumulate with increasing time steps.

Table 3 presents the maximum forecast time under different MAE thresholds and input window sizes. It can be observed that, under the same MAE threshold, a larger input window size corresponds to a longer maximum forecast time. Similarly, under the same input window size, a larger MAE threshold results in a longer maximum forecast time. When the MAE threshold is set at 0.6°C and the input window size is 90 d, DSTGN has a maximum forecast time 37 d longer than Graph WaveNet, and DSTGN also performs better in other cases.

4.2 Correlation studies

As we show in Fig. 2, the correlation between the different points has been dynamically changing over time due to the complex interaction of multiple factors that affect ocean temperature. To visualize this correlation, we sampled data from Points A, B,

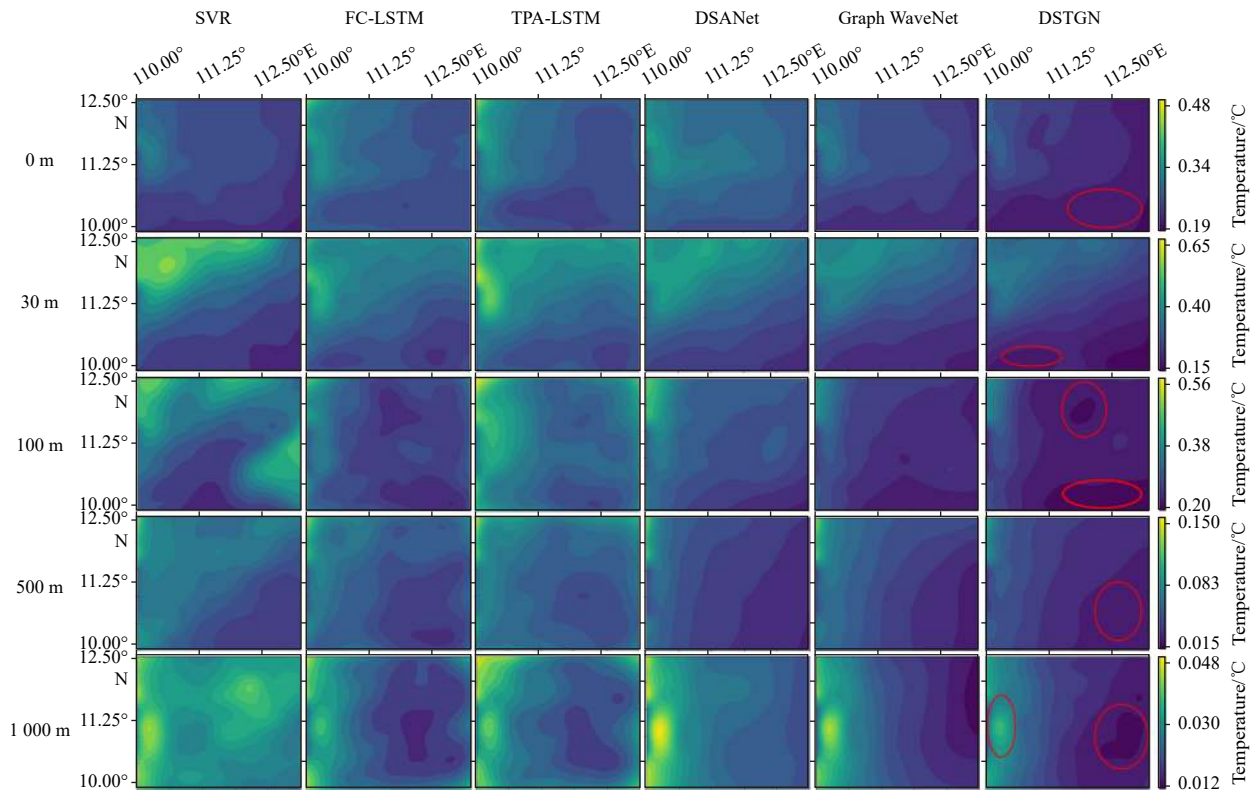


Fig. 9. Prediction error distributions of different models in different ocean temperature layers. The error plots of SVR, FC-LSTM, TPA-LSTM, DSANet, Graph WaveNet, and DSTGN are 0 m, 30 m, 100 m, 500 m, and 1000 m under the sea from top to bottom. The color bar on the right represents the MAE value. The red circle area indicates that our model is significantly better than other models.

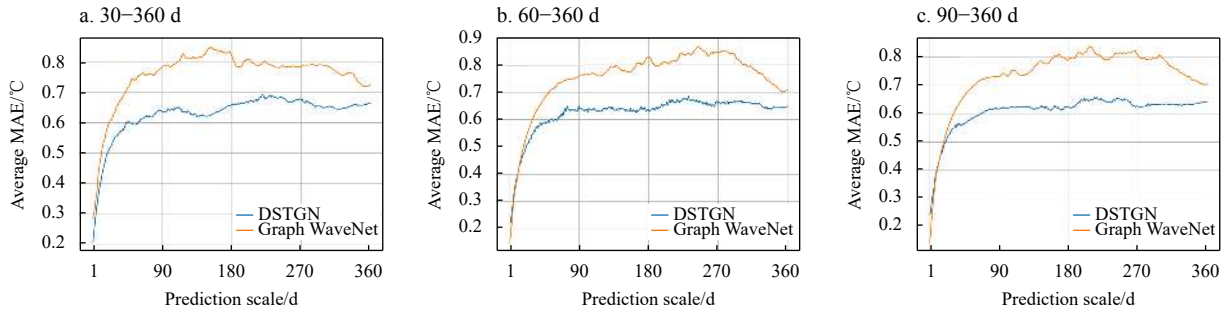


Fig. 10. Prediction average MAE curves in different input windows.

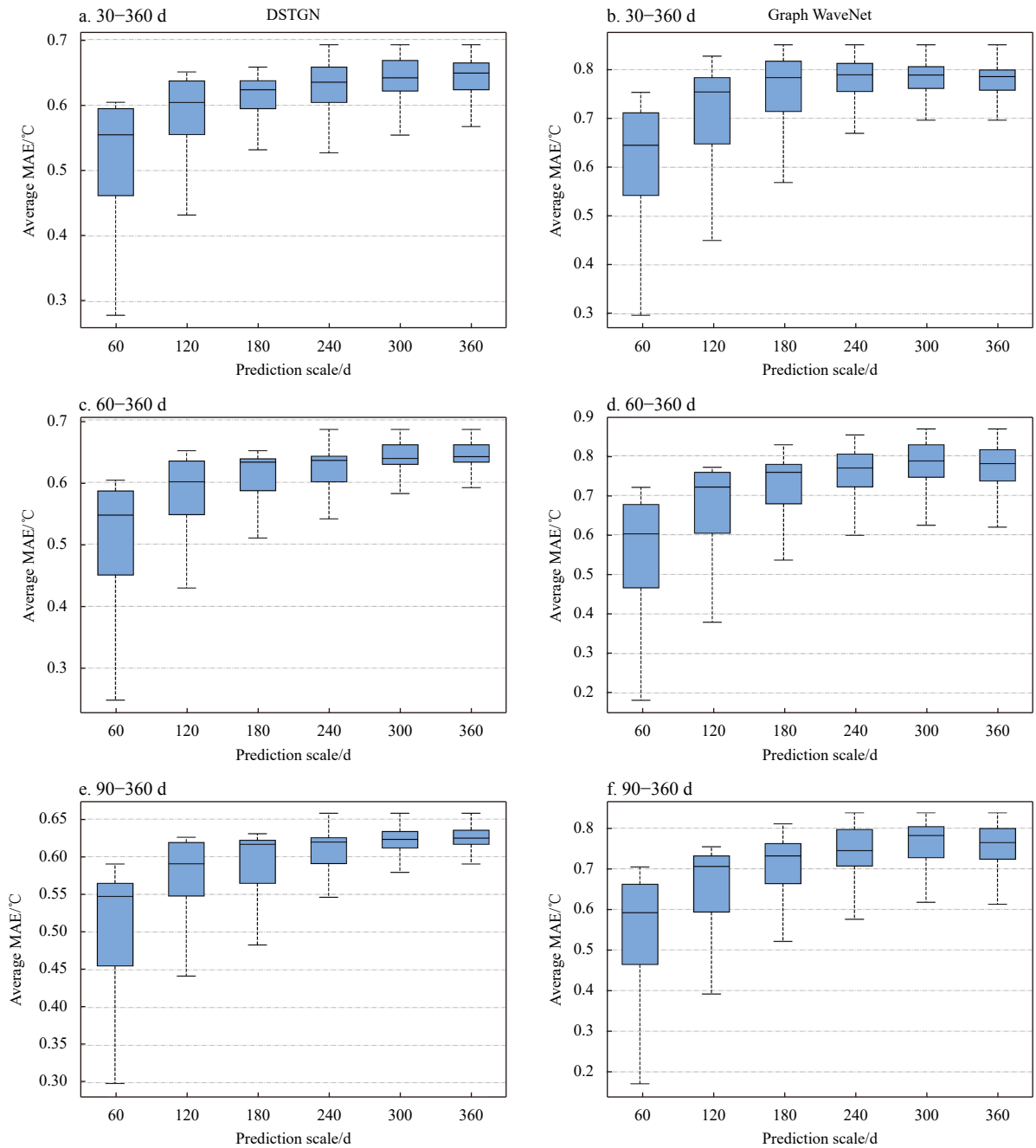


Fig. 11. Boxplots of prediction average MAE with different output windows and input windows.

and C (Fig. 1) on the test set using a 30-d sliding window, calculated and visualized the Pearson correlation coefficient between them two by two, as shown in Fig. 12.

As we know, the Pearson correlation coefficient measures the strength of the linear relationship between two variables and has a value between -1 and 1 . Positive values indicate a positive correlation, the closer it is to 1 , the stronger the positive correlation and vice versa. As can be seen from the green curve in Fig. 12, the correlation between the three points two by two has changed considerably over time and does not have a clear pattern overall, with correlations still varying relatively widely even in the same month in different years. This may be because our window size is relatively small and the effect on ocean temperature is more significant and uncontrollable for short-term data. But even in this case, the correlation between DSTGN's predictions (red curve) still fits the true correlation very well, which shows that DSTGN is able to capture the variability of this correlation relatively well.

Table 3. Maximum predictable time with different MAE thresholds

Window	Model	MAE/ $^{\circ}\text{C}$				
		0.3	0.4	0.5	0.6	0.7
30–360 d	DSTGN	4	10	19	44	/
	Graph WaveNet	2	7	12	21	43
60–360 d	DSTGN	5	11	22	53	/
	Graph WaveNet	5	11	18	30	51
90–360 d	DSTGN	4	9	20	67	/
	Graph WaveNet	5	10	19	31	57

Note: / denote no data.

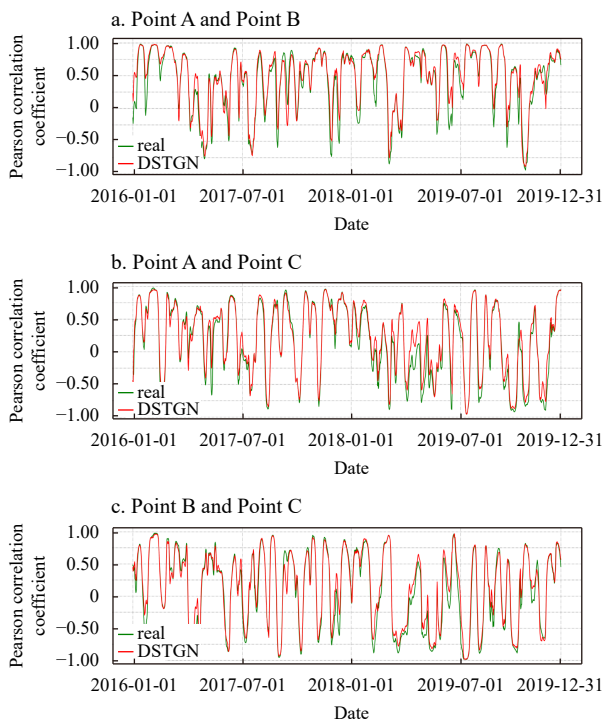


Fig. 12. Pearson correlation coefficients for Points A, B, and C (shown in Fig. 1). The time range is from 1 January 2016 to 31 December 2019, the green curve represents the Pearson correlation coefficient for the true temperature and the red curve represents the Pearson correlation coefficient for the predicted values of the DSTGN. Data format: YYYY-MM-DD.

4.3 Limitations of DSTGN

Although DSTGN outperforms other models in predicting 3D-OT, there are still some limitations. Firstly, As can be seen from Fig. 3, DSTGN requires constant updating of the graph structure to make the dynamic graph structure learned by the model more consistent with the true correlations contained in the dataset, which leads to a relatively high time complexity of DSTGN, which can become computationally expensive when the dynamic graph is very large, and therefore further optimization of the algorithm is required to reduce the computational cost. Secondly, ocean temperature is influenced by multiple factors, such as solar radiation, atmospheric circulation, and surface winds. However, DSTGN can currently only handle single element data and cannot capture the interactions between different features, limiting its application in complex environments. Finally, DSTGN requires complete time-series data for prediction, but in practical applications, ocean temperature data often contain noise and missing values, which may affect the model's prediction performance.

5 Conclusions

In this paper, we propose a novel spatiotemporal graph neural network for predicting 3D-OT. Compared to SST prediction, 3D-OT involves sea temperature layers of different depths, and the dependencies between data are more complex and difficult to predict accurately. From the perspective of data, we found that the dependencies between 3D-OT data change dynamically over time, and the model proposed in this paper can automatically capture both types of dependencies without a priori knowledge, which expands the application scenario of the GNN and plays an important role in predicting and obtaining unknown relationships.

Longer-term future 3D-OT prediction is still in its infancy, and our work is creative in terms of using dynamic graph neural networks for 3D-OT prediction. In addition, our model is more general because it can handle time-series data with or without predefined graphs. In the future, our model is not limited to 3D-OT prediction and can be better applied to datasets with more complex environments, larger fluctuation characteristics, and stronger spatiotemporal correlation.

References

- Aparna S G, D'souza S, Arjun N B. 2018. Prediction of daily sea surface temperature using artificial neural networks. *International Journal of Remote Sensing*, 39(12): 4214–4231, doi: [10.1080/01431161.2018.1454623](https://doi.org/10.1080/01431161.2018.1454623)
- Barnston A G, Tippett M K, Ranganathan M, et al. 2019. Deterministic skill of ENSO predictions from the North American Multimodel Ensemble. *Climate Dynamics*, 53(12): 7215–7234, doi: [10.1007/s00382-017-3603-3](https://doi.org/10.1007/s00382-017-3603-3)
- Collins D C, Reason C J C, Tangang F. 2004. Predictability of Indian Ocean sea surface temperature using canonical correlation analysis. *Climate Dynamics*, 22(5): 481–497, doi: [10.1007/s00382-004-0390-4](https://doi.org/10.1007/s00382-004-0390-4)
- Dong Yihe, Cordonnier J B, Loukas A. 2021. Attention is not *all* you need: Pure attention loses rank doubly exponentially with depth. In: *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2793–2803
- Gao Ziheng, Li Zhuolin, Yu Jie, et al. 2023. Global spatiotemporal graph attention network for sea surface temperature prediction. *IEEE Geoscience and Remote Sensing Letters*, 20: 1500905
- Garcia-Gorriç E, Garcia-Sanchez J. 2007. Prediction of sea surface temperatures in the western Mediterranean Sea by neural networks using satellite observations. *Geophysical Research Letters*, 34(11): L11603
- Glorot X, Bordes A, Bengio Y. 2011. Deep sparse rectifier neural net-

- works. In: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. Fort Lauderdale, 315–323
- Guo Shengnan, Lin Youfang, Feng Ning, et al. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence. Honolulu: AAAI Press, 922–929
- He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 770–778
- Kipf T N, Welling M. 2017. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv: 1609.02907
- Krishnamurti T N, Chakraborty A, Krishnamurti R, et al. 2006. Seasonal prediction of sea surface temperature anomalies using a suite of 13 coupled atmosphere-ocean models. *Journal of Climate*, 19(23): 6069–6088, doi: [10.1175/JCLI3938.1](https://doi.org/10.1175/JCLI3938.1)
- Li Zhuolin, Yu Jie, Zhang Xiaolin, et al. 2022. A multi-hierarchical attention-based prediction method on time series with spatio-temporal context among variables. *Physica A: Statistical Mechanics and its Applications*, 602: 127664, doi: [10.1016/j.physa.2022.127664](https://doi.org/10.1016/j.physa.2022.127664)
- Lins I D, Araujo M, das Chagas Moura M, et al. 2013. Prediction of sea surface temperature in the tropical Atlantic by support vector machines. *Computational Statistics & Data Analysis*, 61: 187–198
- Liu Meng, Gao Hongyang, Ji Shuiwang. 2020. Towards deeper graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. USA: ACM, 338–348
- Luo Jingjia, Masson S, Behera S, et al. 2005. Seasonal climate predictability in a coupled OAGCM using a different approach for ensemble forecasts. *Journal of Climate*, 18(21): 4474–4497, doi: [10.1175/JCLI3526.1](https://doi.org/10.1175/JCLI3526.1)
- Mendoza V M, Villanueva E E, Adem J. 1997. Numerical experiments on the prediction of sea surface temperature anomalies in the Gulf of Mexico. *Journal of marine systems*, 13(1–4): 83–99, doi: [10.1016/S0924-7963\(96\)00120-0](https://doi.org/10.1016/S0924-7963(96)00120-0)
- Neelin J D. 1990. A hybrid coupled general circulation model for El Niño studies. *Journal of the Atmospheric Sciences*, 47(5): 674–693, doi: [10.1175/1520-0469\(1990\)047<0674:AHGCGM>2.0.CO;2](https://doi.org/10.1175/1520-0469(1990)047<0674:AHGCGM>2.0.CO;2)
- Patil K, Deo M C, Ghosh S, et al. 2013. Predicting sea surface temperatures in the North Indian Ocean with nonlinear autoregressive neural networks. *International Journal of Oceanography*, 2013: 302479
- Patil K, Deo M C, Ravichandran M. 2016. Prediction of sea surface temperature by combining numerical and neural techniques. *Journal of Atmospheric and Oceanic Technology*, 33(8): 1715–1726, doi: [10.1175/JTECH-D-15-0213.1](https://doi.org/10.1175/JTECH-D-15-0213.1)
- Shi Lei, Zhang Yofan, Cheng Jian, et al. 2019. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 12026–12035
- Solanki H U, Bhatpuria D, Chauhan P. 2015. Signature analysis of satellite derived SSHa, SST and chlorophyll concentration and their linkage with marine fishery resources. *Journal of Marine Systems*, 150: 12–21, doi: [10.1016/j.jmarsys.2015.05.004](https://doi.org/10.1016/j.jmarsys.2015.05.004)
- Stockdale T N, Balmaseda M A, Vidard A. 2006. Tropical Atlantic SST prediction with coupled ocean-atmosphere GCMs. *Journal of Climate*, 19(23): 6047–6061, doi: [10.1175/JCLI3947.1](https://doi.org/10.1175/JCLI3947.1)
- Sumner M D, Michael K J, Bradshaw C J A, et al. 2003. Remote sensing of Southern Ocean sea surface temperature: Implications for marine biophysical models. *Remote Sensing of Environment*, 84(2): 161–173, doi: [10.1016/S0034-4257\(02\)00103-7](https://doi.org/10.1016/S0034-4257(02)00103-7)
- Sun Yongjiao, Yao Xin, Bi Xin, et al. 2021. Time-series graph network for sea surface temperature prediction. *Big Data Research*, 25: 100237, doi: [10.1016/j.bdr.2021.100237](https://doi.org/10.1016/j.bdr.2021.100237)
- Sun Weifu, Zhang Jie, Meng Junmin, et al. 2019. Sea surface temperature characteristics and trends in China offshore seas from 1982 to 2017. *Journal of Coastal Research*, 90(SI): 27–34
- Szegedy C, Liu Wei, Jia Yangqing, et al. 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston: IEEE, 1–9
- Vaswani A, Shazeer N, Parmar N, et al. 2017. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc
- Wang Tingting, Li Zhuolin, Geng Xiulin, et al. 2022. Time series prediction of sea surface temperature based on an adaptive graph learning neural model. *Future Internet*, 14(6): 171, doi: [10.3390/fi14060171](https://doi.org/10.3390/fi14060171)
- Wentz F J, Gentemann C, Smith D, et al. 2000. Satellite measurements of sea surface temperature through clouds. *Science*, 288(5467): 847–850, doi: [10.1126/science.288.5467.847](https://doi.org/10.1126/science.288.5467.847)
- Wu Zonghan, Pan Shirui, Long Guodong, et al. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. USA: ACM, 753–763
- Xiao Changjiang, Chen Nengcheng, Hu Chuli, et al. 2019. A spatiotemporal deep learning model for sea surface temperature field prediction using time-series satellite data. *Environmental Modelling & Software*, 120: 104502
- Xiao Lin, Shi Jian, Jiang Guorong, et al. 2018. The influence of ocean waves on sea surface current field and sea surface temperature under the typhoon background. *Marine Science Bulletin (in Chinese)*, 37(4): 396–403
- Xie J, Zhang J Y, Yu J, et al. 2020. An adaptive scale sea surface temperature predicting method based on deep learning with attention mechanism. *IEEE Geoscience and Remote Sensing Letters*, 17(5): 740–744, doi: [10.1109/LGRS.2019.2931728](https://doi.org/10.1109/LGRS.2019.2931728)
- Xiong Ruibin, Yang Yunchang, He Di, et al. 2020. On layer normalization in the transformer architecture. In: Proceedings of the 37th International Conference on Machine Learning. JMLR. org, 10524–10533
- Xue Yan, Leetmaa A. 2000. Forecasts of tropical Pacific SST and sea level using a Markov model. *Geophysical Research Letters*, 27(17): 2701–2704, doi: [10.1029/1999GL011107](https://doi.org/10.1029/1999GL011107)
- Yang Yuting, Dong Junyu, Sun Xin, et al. 2018. A CFCC-LSTM model for sea surface temperature prediction. *IEEE Geoscience and Remote Sensing Letters*, 15(2): 207–211, doi: [10.1109/LGRS.2017.2780843](https://doi.org/10.1109/LGRS.2017.2780843)
- Zhang Kun, Geng Xupu, Yan Xiaohai. 2020a. Prediction of 3-D ocean temperature by multilayer convolutional LSTM. *IEEE Geoscience and Remote Sensing Letters*, 17(8): 1303–1307, doi: [10.1109/LGRS.2019.2947170](https://doi.org/10.1109/LGRS.2019.2947170)
- Zhang Xiaoyu, Li Yongqing, Frery A C, et al. 2022. Sea surface temperature prediction with memory graph convolutional networks. *IEEE Geoscience and Remote Sensing Letters*, 19: 8017105
- Zhang Zhen, Pan Xinliang, Jiang Tao, et al. 2020b. Monthly and quarterly sea surface temperature prediction based on gated recurrent unit neural network. *Journal of Marine Science and Engineering*, 8(4): 249, doi: [10.3390/jmse8040249](https://doi.org/10.3390/jmse8040249)
- Zhang Qin, Wang Hui, Dong Junyu, et al. 2017. Prediction of sea surface temperature using long short-term memory. *IEEE Geoscience and Remote Sensing Letters*, 14(10): 1745–1749, doi: [10.1109/LGRS.2017.2733548](https://doi.org/10.1109/LGRS.2017.2733548)
- Zuo Xinyi, Zhou Xiaofeng, Guo Daquan, et al. 2022. Ocean temperature prediction based on stereo spatial and temporal 4-D convolutional model. *IEEE Geoscience and Remote Sensing Letters*, 19: 1003405