# Dual-Branched Spatio-Temporal Fusion Network for Multihorizon Tropical Cyclone Track Forecast

Zili Liu, Kun Hao, Xiaoyi Geng, Zhengxia Zou, and Zhenwei Shi, *Member, IEEE*

***Abstract*—A tropical cyclone (TC) is a typical extreme tropical weather system, which could cause serious disasters in transit areas. Accurate TC track forecasting is the key to reducing casualties and damages, however, long-term forecasting of TCs is a challenging problem due to their extremely high dynamics and uncertainty. Existing TC track forecasting methods mainly focus on utilizing a single modality of source data, meanwhile, suffer from limited long-term forecasting capability and high computational complexity. In this article, we propose to address the abovementioned challenges from a new perspective—by utilizing large-scale spatio-temporal multimodal historical data and advanced deep learning techniques. A novel multihorizon TC track forecasting model named dual-branched spatio-temporal fusion network (DBF-Net) is proposed and evaluated. DBF-Net contains a TC features branch that extracts temporal features from 2-D state vectors and a pressure field branch that extracts spatio-temporal features from reanalysis 3-D pressure field. We show that with the abovementioned design, DBF-Net can fully exploit the implicit associations of multimodal data, achieving advantages that unimodal data-based method does not have. Extensive experiments on 39 years of historical TCs track data in the Northwest Pacific show that our DBF-Net achieves significant accuracy improvement compared with previous TCs track forecast methods.**

***Index Terms*—Multimodal data fusion, spatio-temporal data, tracking forecast, tropical cyclones (TCs).**

## I. Introduction

TROPICAL cyclones (TCs, a.k.a, typhoons or hurricanes) are low-pressure vortexes occurring over the tropical or subtropical oceans. TCs are one of the major meteorological disasters facing mankind. Accurate forecasting for the TCs trajectory can greatly reduce the casualties and property damages caused.

The research on TCs track forecast has gone through four stages since the 1960s—empirical methods, statistical methods, numerical methods, and deep learning methods. Early methods of TCs track forecast were limited by observation techniques and computational devices and could rely only on the subjective experience for achieving forecast. Thus, some traditional methods, such as extrapolation and similar path methods were developed [1]. From the 1980s, with the rapid development of statistical models, forecasting models based on statistical regression methods, such as climatology and persistence (CLIPER) [2] were proposed. However, limited representation capabilities of manual feature selection make it difficult to produce accurate forecast results. Since the 1990s, with the improvement of observation techniques and computer performance, numerical weather prediction (NWP) systems (e.g., American National Hurricane Center Track and Intensity Model) gradually become the mainstream choice for official meteorological forecasting agencies. NWP achieves forecasting by solving complex PDEs of weather dynamics. However, NWP is very computationally expensive and requires the support of supercomputer platforms. In recent years, machine learning especially deep learning has developed rapidly. Various deep neural networks (DNNs) based on deep learning have revolutionized real-world applications such as computer vision [3], natural language processing [4], and time series forecasting [5]. Since the computational complexity of DNNs is much smaller than that of traditional NWP models, many DNN variants have been proposed recently to predict TCs track [6]–[14]. In this article, we also focus on deep learning-based TC track forecasting.

Long-term forecasting of TCs is a challenging problem due to their extremely high dynamics and uncertainty. The trajectory of a TC could be affected by various physical quantities in the atmosphere and ocean, such as pressure field, wind field, and sea surface temperature. Existing deep learning-based TC track forecasting methods mainly focus on utilizing a single modality of source data [7]–[11], [14], meanwhile, suffer from limited long-term forecasting capability and high computational complexity. In this article, we propose to address the abovementioned challenges from a new perspective—by utilizing large-scale spatio-temporal multimodal historical data and advanced deep learning techniques. We define the source data into the following three categories: *inherent features of TCs*, *remote sensing images,* and *meteorological fields*. The key to deep learning-based forecasting methods is the full exploitation of different types of data.

The *inherent features of TCs* at a particular time are always represented by a column vector or tensor, which contains information, such as the latitude, longitude, and intensity of the

Manuscript received March 14, 2022; revised April 21, 2022; accepted April 22, 2022. Date of publication April 26, 2022; date of current version May 20, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62125102. *(Corresponding author: Zhenwei Shi.)*

Zili Liu, Kun Hao, Xiaoyi Geng, and Zhenwei Shi are with Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China, and with the Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, China, and also with the State Key Laboratory of Virtual Reality Technology and Systems, School of Astronautics, Beihang University, Beijing 100191, China (e-mail: liuzili@buaa.edu.cn; haokun@buaa.edu.cn; gxy0809@buaa.edu.cn; shizhenwei@buaa.edu.cn).

Zhengxia Zou is with the Department of Guidance, Navigation and Control, School of Astronautics, Beihang University, Beijing 100191, China (e-mail: zhengxiazou@buaa.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2022.3170299

center of TCs at that time. Infantile deep learning-based TCs track forecast model mainly utilizes historical inherent features of TCs to predict the future locations of the TCs. The classical multilayer perceptions [7], [8] and various time series prediction models, such as recurrent neural networks (RNNs) [9], [10], long short-term memory (LSTM) model [11], and bidirection gate recurrent unit model [14], are used to learn the time-series pattern of the data. However, the time series models based only on inherent features of TCs usually have limited accuracy of track forecast due to the lack of consideration of factors affecting TCs' trajectory.

Compared with the two-dimensional (2-D) *inherent features of TCs* vector, the 3-D *remote sensing images*, and *meteorological fields* can describe the relevant information around the TCs. As for the TCs track forecasting using remote sensing images, it can be treated as a special kind of video frame prediction task. Rüttgers *et al.* [15] used a generative adversarial network (GAN) to predict the TCs track images and the corresponding location of TCs center. Wu *et al.* [16] proposed a multitask machine learning framework based on an improved GAN to predict the track and intensity of TCs simultaneously. The track forecast methods mentioned above take full advantage of the powerful performance of GANs in the field of computer vision. However, the remote sensing images used in such methods need to be acquired from geostationary satellites to ensure high temporal resolution, and the images cannot represent the physical factors affecting the TCs trajectory.

*Meteorological fields*, such as pressure fields and wind fields, are the main factors affecting the trajectory of TCs. In 2017, Mudigonda *et al.* [17] proposed the CNN-LSTM model for segmenting and tracking TCs and verified the direct high correlation of TC tracks in meteorological fields. Kim [12] proposed a convolutional LSTM-based spatio-temporal model that predicts the trajectory map based on the density map sequence generated from the wind velocity and precipitation fields. But the predicted trajectory map cannot reflect the exact location of the TCs precisely. Therefore, how to efficiently fuse the *meteorological fields* data into the TCs track forecast model to improve the forecast accuracy has gradually become the mainstream research direction in recent years [13], [18]. Due to the large variation in the distribution of different meteorological fields, Giffard-Roisin *et al.* [13] used different CNN models to encode the reanalysis data of wind and pressure field, respectively, and fuse them with past track data of TCs. However, multiple CNN models increase the number of parameters and computational complexity of the forecast model, and the model is difficult to train. In addition, the inclusion of excessive use of meteorological field data weakens the role of the inherent features data of TCs and does not adequately learn the time-series features of the data. Therefore, how to efficiently utilize the meteorological field data and fully exploit the intrinsic time-series information of the inherent features of TCs still needs further research.

To solve the problems mentioned above, this article tries to exploit the temporal information in the inherent features data of TCs and the spatio-temporal information in reanalysis 3-D pressure field data, and proposes a dual-branched spatio-temporal fusion network (DBF-Net) for multihorizon TC track forecast

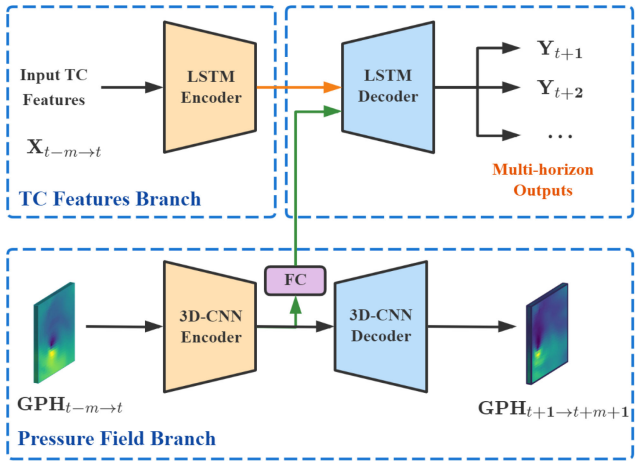



Fig. 1. Overview of the proposed DBF-Net for multihorizon TC track forecast.

(i.e., predicting the TCs' track at multiple future time steps [19]). Specifically, as shown in Fig. 1, the time-series features of the input TC features are extract by an LSTM-based [20] network in the TC features branch efficiently. Meanwhile, the spatio-temporal features of the geopotential height (GPH) around TCs are encoded by the 3D-CNN-based [21] network in the pressure field branch, and fused into the first branch to complement the track forecasting information by predicting the GPH at multiple future time steps, and provide the multihorizon TCs trajectory forecasting outputs.

Through efficient spatio-temporal feature extraction and fusion of the two types of data, the 24 h forecast accuracy of DBF-Net on historical TC tracks data in the Northwest Pacific (WNP) is 119 km, which is much better compared with other deep learning-based methods [13], [14]. Besides, we also compare our method with other traditional methods, such as extrapolation [1], CLIPER model [2], and NWP methods [22]–[25]. Finally, we exhibit the forecast results for several individual cases of TC events for further analysis and verification.

## II. Methodology

In this section, we will introduce the proposed DBF-Net in detail. The overall architecture of the DBF-Net is shown in Fig. 1. The two branches contained in DBF-Net are split into three submodules and will be introduced separately. Before that, the basic LSTM and 3D-CNN modules used in DBF-Net will be briefly introduced.

### A. Preliminaries

We formally introduce symbols and notations in this section. In DBF-Net, there are two types of data as the input, where $\mathcal{X}_t = \{\mathbf{X}_j \in \mathbb{R}^p \mid j \in [t-m, t], j \in \mathbb{Z}\}$ represents the input historical inherent features sequence of TC and $\mathcal{G}_t = \{\mathbf{GPH}_j \in \mathbb{R}^{q\times q} \mid j \in [t-m, t], j \in \mathbb{Z}\}$ represents the input historical reanalysis 3-D GPH data. Given the initial forecast time $t$ and the corresponding input data $\mathcal{X}_t$ and $\mathcal{G}_t$, the output multihorizon

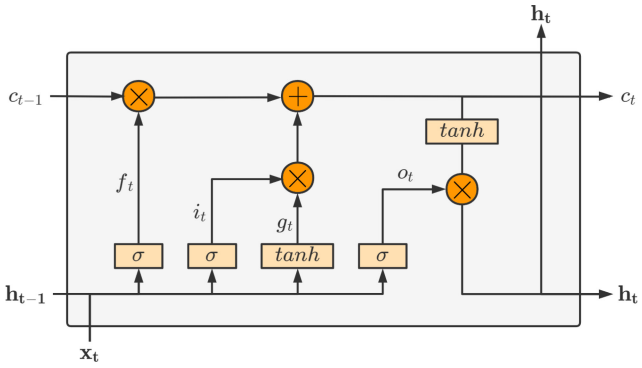



Fig. 2. Structure of LSTM module.

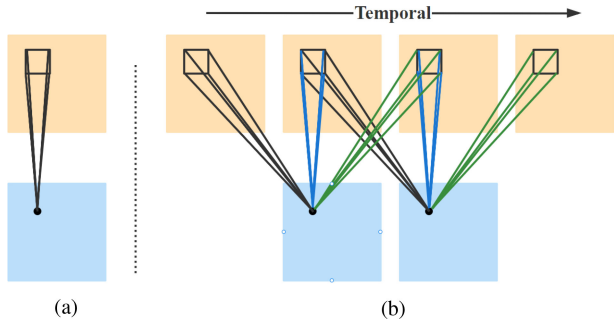



Fig. 3. Difference between (a) 2-D convolution and (b) 3-D convolution. The black boxes represent the convolution kernel and the lines with same color represent the same kernel weight value.

TC track prediction can be computed by

$$\mathcal{Y}_\tau = \mathbb{M}(\mathcal{X}_t, \mathcal{G}_t) \tag{1}$$

where $\mathcal{Y}_\tau = \{\mathbf{Y}_j = (\text{Lat}_{t+j} - \text{Lat}_t, \text{Lon}_{t+j} - \text{Lon}_t) \mid j \in [1, \tau], j \in \mathbb{Z}\}$, $\text{Lat}_t$ and $\text{Lon}_t$ are the latitude and longitude of TC center at time $t$. $\mathbb{M}(\cdot)$ represents the end-to-end DBF-Net. It should be noted that, we use the relative change in latitude and longitude $\mathbf{Y}_j$ as the output of DBF-Net instead of the direct location. The reason for this is that the pressure field data is cropped from the center of TCs and the local information is more suitable for digging the relative changes in TCs motion.

### B. Long Short-Term Memory

RNNs [26] are one of the popular models for capturing the temporal features of sequence data. RNNs consist of multiple neural networks stacked in the time dimension, and the temporal features could be extracted by utilizing both input data $\mathbf{x}_t$ at time $t$ and the extracted latent features $\mathbf{h}_{t-1}$at time $t-1$. However, vanilla RNNs suffer from gradient disappearance and gradient explosion when using long term sequence data for training [27]. Therefore, numerous variants have been proposed to solve the problems mentioned above [20], [28], including the well-known LSTM model [20].

As shown in Fig. 2, the LSTM improves the vanilla RNNs by replacing the traditional ANN module in RNNs with a complex LSTM module. In order to maintain the latent information for both long and short term, namely *cell state*, the following three different gates in LSTM module are proposed for different motivations: forget gate, input gate, and output gate.

Among them, the forget gate controls the remaining information proportion of the cell state $c_{t-1}$ and the proportion can be computed by

$$f_t = \sigma(W_f \cdot [\mathbf{x}_t, \mathbf{h}_{t-1}] + b_f) \tag{2}$$

where $W_f$ and $b_f$ are the weight and bias value in a fully connected layer. $\sigma$ represents the Sigmoid function.

As for the input gate, it determines the amount of the candidate information feeding into the cell state by controlling the proportion index. The candidate information $g_t$ and proportion index $i_t$ can be computed by

$$\begin{aligned} i_t &= \sigma(W_i \cdot [\mathbf{x}_t, \mathbf{h}_{t-1}] + b_i) \\ g_t &= \tanh(W_g \cdot [\mathbf{x}_t, \mathbf{h}_{t-1}] + b_g). \end{aligned} \tag{3}$$

Then, the cell state can be updated from time $t-1$ to time $t$ by

$$c_t = f_t \times c_{t-1} + i_t \times g_t. \tag{4}$$

Lastly, the output gate updates the latent variable by controlling the proportion index of cell state at time $t$, that is

$$\begin{aligned} o_t &= \sigma(W_o \cdot [\mathbf{x}_t, \mathbf{h}_{t-1}] + b_o) \\ \mathbf{h}_t &= o_t \times \tanh(c_t). \end{aligned} \tag{5}$$

Through the operations mentioned above, LSTM could efficiently extract the temporal features by controlling the proportions of different information flow and the cell state to ensure the gradient is always within a proper range, which avoids the gradient disappearance and gradient explosion to some extent.

### C. 3D-CNN

Convolutional neural networks (CNNs) have achieved remarkable progress in extracting spatial features from image-like 2-D data. The local receptive field and shared weights make it possible for CNNs to learn the local spatial relevance of the 2-D data with fewer parameters compared with the traditional fully connected layer. However, for video-like 3-D data, the traditional 2-D convolution can not capture the relevance of the 2-D images at different time steps. Therefore, Ji *et al.* [21] proposed the 3D-CNN for human action recognition and 3D-CNN has become a popular model for extracting spatio-temporal features from video-like 3-D data.

Fig. 3 illustrates the difference between 2-D and 3-D convolutions. As can be seen in 3(b), the convolution kernel matrix in 3D-CNN not only slides on a single 2-D feature map but also moves in the temporal dimension (same color lines in different feature maps in temporal dimension), thus, both spatial and temporal features could be extracted from 3-D data simultaneously.

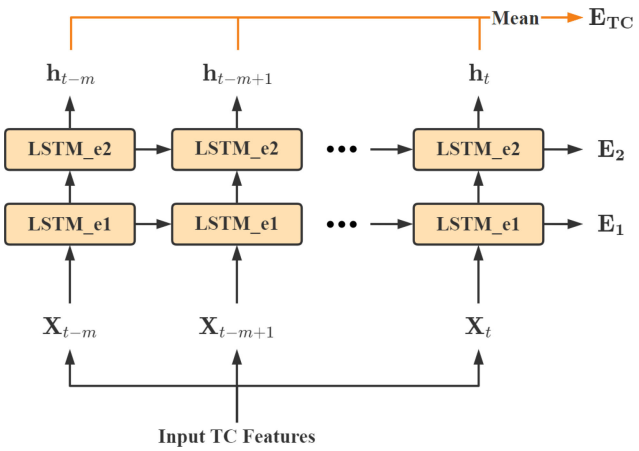



Fig. 4. LSTM-encoder module in TC features branch.

### D. TC Features Encoder Module

The TC features encoder module in the first branch of DBF-Net plays the role of encoding the inherent features of TCs $\mathbf{X}_t = (x_1, x_2, x_3, x_4, x_5, x_6)$ at multiple historical times. Each $x_j$ in $\mathbf{X}_t$ represents the latitude at time $t$, longitude at time $t$, maximum wind speed near the center at the bottom at time $t$, latitude difference between time $t$ and $t-1$, longitude difference between time $t$ and $t-1$, and wind speed difference between time $t$ and $t-4$, respectively. The features mentioned above are also the classical persistence factors in statistical forecasting methods. As a result, the purpose of the TC features encoder module is to encode time series features of the persistence factors.

As shown in Fig. 4, the TC features encoder module consists of a two-layer stacked LSTM encoder. Each $\mathbf{X}_t$ in sequence $\mathcal{X}_t$ is passed sequentially into the two-layer LSTM encoding module and produces the latent variable $\mathbf{h}_t$. The output of each LSTM encoding layer is fed into the corresponding layer at next time step. The specific operation procedure in each LSTM encoding layer is as follows:

$$i_t = \sigma(W_{ii}\mathbf{X}_t + b_{ii} + W_{hi}\mathbf{h}_{t-1} + b_{hi}) \tag{6}$$

$$f_t = \sigma(W_{if}\mathbf{X}_t + b_{if} + W_{hf}\mathbf{h}_{t-1} + b_{hf}) \tag{7}$$

$$g_t = \tanh(W_{ig}\mathbf{X}_t + b_{ig} + W_{hg}\mathbf{h}_{t-1} + b_{hg}) \tag{8}$$

$$c_t = f_t \times c_{t-1} + i_t \times g_t \tag{9}$$

$$o_t = \sigma(W_{io}\mathbf{X}_t + b_{io} + W_{ho}\mathbf{h}_{t-1} + b_{ho}) \tag{10}$$

$$\mathbf{h}_t = o_t \times \tanh(c_t) \tag{11}$$

where the $i_t$, $f_t$, and $o_t$ represent the output of the input gate, forget gate, and output gate in LSTM model. $W_{ii}(W_{hi})$, $W_{if}(W_{hf})$, and $W_{io}(W_{ho})$ are the corresponding weight matrix related to the $\mathbf{X}_t(\mathbf{h}_{t-1})$. $c_t$ is the cell state fed into the next time step together with the latent variable $\mathbf{h}_t$. $\sigma(\cdot)$ is the Sigmoid function.

Given the input sequence $\mathcal{X}_t$ of length $m+1$ and the corresponding latent variable sequence $\{\mathbf{h}_{t-m}, \mathbf{h}_{t-m+1}, \ldots, \mathbf{h}_t\}$, $t$ is the initial forecast time, we can compute the final time series code of the TC features by

$$\mathbf{E_{TC}} = \frac{1}{m+1}\sum_{i=0}^{m}\mathbf{h}_{t-i}. \tag{12}$$

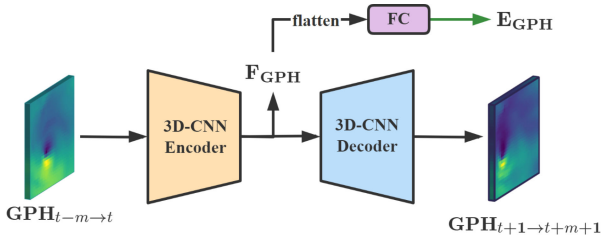



Fig. 5. Pressure field branch architecture.

TABLE I
3D-CNN-BASED ENCODER ARCHITECTURE OF THE PRESSURE FIELD BRANCH

| Layers | Kernel Size | Stride | In Channel |
|---|---|---|---|
| Conv_1 | $3\times3\times3$ | $1\times1\times1$ | 1 |
| MaxPool_1 | $1\times2\times2$ | $1\times2\times2$ | 16 |
| Conv_2 | $3\times3\times3$ | $1\times1\times1$ | 16 |
| MaxPool_2 | $1\times2\times2$ | $1\times2\times2$ | 32 |
| Conv_3 | $3\times3$ | $1\times1$ | 32 |
| MaxPool_3 | $2\times2$ | $2\times2$ | 64 |

### E. Pressure Field Branch

To efficiently use the meteorological fields in the vicinity of TCs and improve the forecast accuracy, a 3D-CNN-based encoder–decoder networks are utilized to generate high-level spatio-temporal features from the reanalysis 3-D GPH data at multiple time steps (as shown in Fig. 5).

As shown in Table I, the encoder of the pressure field branch contains three-convolutional layers, the first two of which are 3D-CNN with kernel size $3\times3\times3$. To ensure that the GPH field data cover the full spatial extent that may affect the TC tracks, the window size at each historical time step $\mathbf{GPH}_t$ of input 3-D field $\mathcal{G}_t$ is set to $51\times51$ values, which is approximately a radius of 1400 km (the resolution of the reanalysis GPH data is 0.5°). We choose LeakyReLU as the activation function of the encoder to enhance the nonlinear representation of the model. The output high-level spatio-temporal features can be computed by

$$\begin{aligned}\mathbf{E_{GPH}} &= \mathrm{FC}(\mathrm{flatten}(\mathbf{F_{GPH}}))\\ &= \mathrm{FC}(\mathrm{flatten}(\mathrm{Encoder}(\mathcal{G}_t)))\end{aligned} \tag{13}$$

where $\mathrm{FC}(\cdot)$ is a fully connected layer. $\mathrm{flatten}(\cdot)$ is the flatten operation that flattening output feature map of the 3D-CNN encoder.

As for the decoder in the pressure field branch, its structure is symmetrical with the 3D-CNN encoder and the transpose convolution is used to recover the spatio-temporal information from high-level features. It predicts the future $m+1$ time steps of the GPH, which is same length of time as the input $\mathcal{G}_t$.

The loss function of the pressure field branch is computed by

$$L_{\text{GPH}} = \sum_{i=t+1}^{t+m+1} ||\text{Decoder}(\mathbf{F_{GPH}}) - \mathbf{TGPH}_i||_1 = \sum_{i=t+1}^{t+m+1} ||\mathbf{GPH}_i - \mathbf{TGPH}_i||_1 \quad (14)$$

where $\mathbf{TGPH}_i$ is the target value of the future GPH data. $||\cdot||_1$ is the $l_1$-norm function.

### F. Dual-Branched Features Fusion Decoder Module

Based on the LSTM-based encoder in TC features branch and the pressure field branch mentioned above, there are three types of intermediate variables from two branches that fed into the LSTM-based decoder module that is the final time series code of the TC features $\mathbf{E_{TC}}$, the high-level spatio-temporal reanalysis 3-D GPH features $\mathbf{E_{GPH}}$ and the output features from LSTM encoder layers $\mathbf{E}_1$ and $\mathbf{E}_2$ (as shown in Fig. 6). In this section, we will introduce the LSTM-based decoder module for efficiently fusing the dual-branched multimodal features and generating multihorizon TC track forecasting results.

The proposed LSTM-based decoder module contains a two-layers stacked LSTM decoder and the subsequent two fully connected layers. The detailed decoding and feature fusion process is defined as follows:

$$i_t = \sigma(W_{Ti}\mathbf{E_{TC}} + W_{Gi}\mathbf{E_{GPH}} + W_{Yi}\mathbf{Y}_{t-1} + W_{hi}\mathbf{h}_{t-1} + b_i) \quad (15)$$

$$f_t = \sigma(W_{Tf}\mathbf{E_{TC}} + W_{Gf}\mathbf{E_{GPH}} + W_{Yf}\mathbf{Y}_{t-1} + W_{hf}\mathbf{h}_{t-1} + b_f) \quad (16)$$

$$g_t = (W_{Tg}\mathbf{E_{TC}} + W_{Gg}\mathbf{E_{GPH}} + W_{Yg}\mathbf{Y}_{t-1} + W_{hg}\mathbf{h}_{t-1} + b_g) \quad (17)$$

$$c_t = f_t \times c_{t-1} + i_t \times g_t \quad (18)$$

$$o_t = \sigma(W_{To}\mathbf{E_{TC}} + W_{Go}\mathbf{E_{GPH}} + W_{Yo}\mathbf{Y}_{t-1} + W_{ho}\mathbf{h}_{t-1} + b_o) \quad (19)$$

$$\mathbf{h}_t = o_t \times \tanh(c_t) \quad (20)$$

$$\mathbf{Y}_t = \text{FC}_2(\text{ReLU}(\text{FC}_1(\mathbf{h}_t))) \quad (21)$$

where the initial state $\mathbf{h}_t$ and $c_t$ are the elements of $\mathbf{E}_1$ and $\mathbf{E}_2$, which are the final state of the LSTM encoder layers. The initial input of the LSTM decoder $\mathbf{Y}_t$ is set to zero. The loss function of the LSTM-based decoder is also $l_1$-norm function, which is defined as follows:

$$L_{\text{loc}} = \sum_{i=t+1}^{t+\tau} ||\text{Decoder}(\mathbf{E_{TC}}, \mathbf{E_{GPH}}, \mathbf{E}_1, \mathbf{E}_2, \mathbf{Y}_t) - \mathbf{Y}_i||_1 = \sum_{i=t+1}^{t+\tau} ||\mathbf{T}_i - \mathbf{Y}_i||_1 \quad (22)$$

where $\mathbf{T}_i$ is the ground-truth changes of latitude and longitude. With the operation mentioned above, the features from both inherent TC features and reanalysis 3-D pressure field can be fused effectively and we can achieve the multihorizon TC track forecasting results $\mathcal{Y}_t$ based on the fused multimodal features.

TABLE II
EXAMPLES OF CMA-BST DATA

| YYYYMMDDHH | I | LAT | LON | PRES | WND | OWD |
|---|---|---|---|---|---|---|
| 1953061506 | 0 | 125 | 1116 | 1000 | 10 | 15 |
| 1953061512 | 0 | 132 | 1117 | 1000 | 10 | 15 |
| 1953061518 | 0 | 142 | 1117 | 1000 | 10 | 15 |
| 1953061600 | 0 | 150 | 1117 | 1000 | 10 | 20 |
| 1953061606 | 0 | 159 | 1112 | 999 | 10 | 20 |

I stands for the intensity level of TCs. LAT and LON are the latitude and longitude of TCs' centers (unit: $\times 0.1^\circ$). PRES stands for the central minimum pressure (unit: hPa). WND and OWD stand for the 2-min maximum and average near-center wind speed, respectively (unit: m/s).

### G. Loss Functions and Multistages Training

We trained our proposed DBF-Net in a three stages manner. First, we only train the TC features encoder module by adding a fully connected layer to directly predict the target value of the TC track and get the pretrained LSTM encoder in the TC features branch. Then, we utilize the reanalysis 3-D pressure fields GPH data to train the pressure field branch of the DBF-Net and learn the temporal dynamic changing of GPH data. Finally, we add the LSTM decoder module into the training pipeline and train the DBF-Net in an end-to-end manner. The loss function at the final step is as follows:

$$L_{\text{final}} = L_{\text{loc}} + \alpha L_{\text{GPH}} + \beta L_2 \quad (23)$$

where the $L_2$ is the regularization term with $l_2$ penalty. $\alpha$ and $\beta$ are hyperparameter. The training schedule detail will be discussed in Section III-C.

## III. EXPERIMENT

In this section, we evaluate our proposed DBF-Net on the historical TC tracks data in WNP. The forecasting performance of DBF-Net is verified by the comparison with other deep learning-based and traditional TC track forecast methods. We also analyze the forecast results for several individual cases of TC events and the specific forecasting characteristics of the DBF-Net.

### A. Dataset

Best track dataset (CMA-BST). The inherent features data of TCs are extracted from the best track (BST) data released by China Meteorological Administration (CMA) [29]. It includes the location and intensity of TCs in the WNP Ocean ($0^\circ$N–$50^\circ$N, $100^\circ$E–$210^\circ$E) at six-hour intervals from 1949 to 2018. Examples of the CMA-BST data are shown in Table II.

GPH dataset (CFSR-GPH). The reanalysis 3-D GPH data in pressure field branch is collected from climate forecast system reanalysis (CFSR) dataset released by The National Centers for Environmental Prediction [30]. CFSR-GPH is grid data with a spatial resolution of $0.5^\circ$ and the temporal resolution is aligned with CMA-BST Since 1979. TCs in WNP are mostly generated

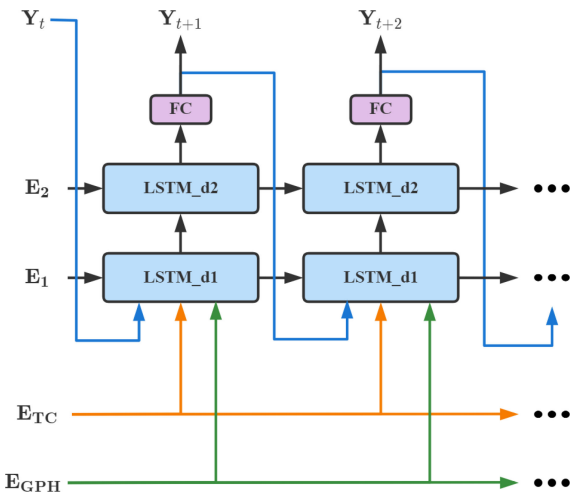



Fig. 6. LSTM-based decoder module in TC features branch with dual-branched features fusion.

TABLE III
DATASET-SPLITTING-BASED ON CMA-BST AND CFSR-GPH

| | Train Set | Val Set | Test Set |
|---|---|---|---|
| Years | 1979-2008 | 2009-2013 | 2014-2018 |
| #TCs | 723 | 104 | 113 |
| #Samples | 13233 | 1847 | 2174 |

at the southern edge of the subtropical high pressure and move along its periphery. Therefore, the 500 hPa geopotential height data is chosen as the background pressure field to describe the activity of TCs.

*Data preprocessing:* For both input data from CMA-BST and CFSR-GPH, we use the maximum–minimum normalization for data preprocessing and feed them into the TC features and pressure field branch, respectively.

*Dataset split:* Based on the CMA-BST and CFSR-GPH dataset mentioned above. We choose overlap of the two datasets, i.e., TCs from 1979 to 2018. And we only keep the TCs with a life cycle greater than four days to ensure the persistence. There are 940 TCs left in this dataset. We make 17 000+ samples for model training, validating, and testing based on a sliding window of length *input sequence length* + *prediction length* (as shown in Fig. 7 and Table III).

### B. Metrics

We use the mean distance error (MDE) to evaluate the TCs track forecast results. The MDE is a commonly used metric to measure the average distance error between model prediction and ground truth. The MDE can be computed by

$$\begin{aligned}\mathrm{MDE} &= 2 \times R \times \arcsin \\ &\sqrt{\sin^2\left(\frac{\varphi_{\mathrm{pre}} - \varphi_{\mathrm{gt}}}{2}\right) + \cos\varphi_{\mathrm{pre}} \cos\varphi_{\mathrm{gt}} \sin^2\left(\frac{\lambda_{\mathrm{pre}} - \lambda_{\mathrm{gt}}}{2}\right)} \\ &\approx \sqrt{\Delta\mathrm{Lat}^2 + \Delta\mathrm{Lon}^2} \times 110 \end{aligned} \tag{24}$$

where $R \approx 6371$˜km represents the radius of earth. $\varphi_{\mathrm{pre}}$ and $\varphi_{\mathrm{gt}}$ stand for the latitude value of prediction and ground truth. $\lambda_{\mathrm{pre}}$ and $\lambda_{\mathrm{gt}}$ stand for the longitude value of prediction and ground truth.

Besides, the skill score is also the index to evaluate the practical availability of the methods, as follows:

$$\text{skill score} = \frac{e_A - e_B}{e_A} \times 100\% \tag{25}$$

where $e_A$ is the prediction error of CLIPER method and $e_B$ is the error of proposed method.

### C. Implementation Details

We train our proposed DBF-Net in a three stages manner with the Pytorch framework, which has been discussed in Section II-E. We use the RMSProp optimizer and set the initial learning rate to 0.001. The batch size of training set is set to 64. The hyper parameter $\alpha$ and $\beta$ in (19) is set to 1.2 and 0.00001, respectively.

For multihorizon forecasting (i.e., predicting the TCs track at multiple future time steps), the output prediction sequence length of the DBF-Net is 4 and the input sequence length is 5. That is we predict the 6 h, 12 h, 18 h, and 24 h TCs tracks based on the historical data from time $t-5$ (30 h prior) to time $t$ (the current time). We train our DBF-Net on a single NVIDIA GeForce GTX 3090 GPU.

### D. Comparison With Statistical/Deep Learning Forecast Methods

We first compare our proposed DBF-Net with other statistical and deep learning based TCs track forecast methods, including the extrapolation method [1], CLIPER method [2], feature fusion network [13], and recent BiGRU-attn [14]. The extrapolation is a simple traditional TCs track forecast method. It assumes that the direction and speed of TCs movement do not change much, and predicts based on the movement direction and velocity at previous times. CLIPER can be treated as the benchmark of other track forecast methods. It uses correlation analysis to screen climate persistence factors and constructed multivariate linear regression models. In this article, we replace the multivariate linear model with a back propagation (BP) neural network model, which enhancing the nonlinear representation of the CLIPER model. We selected 20 factors with strong correlation from 46 climate persistence factors by Pearson correlation analysis and feed them into the BP neural network.

As shown in Table IV, our proposed DBF-Net outperforms previous works. Specifically, compared with the benchmark forecast method CLIPER, DBF-Net achieves better MDEs for all forecast time steps. That is the skill score with respect to the CLIPER is positive, which demonstrates the practical availability of our method. In addition, our DBF-Net also ourperforms previous deep learning based methods. Compared with FFN [13] that utilizes the wind, pressure fields simultaneously, our DBF-Net achieves better results only based on pressure field. This also shows that our method can better encode the effective features of the input data.

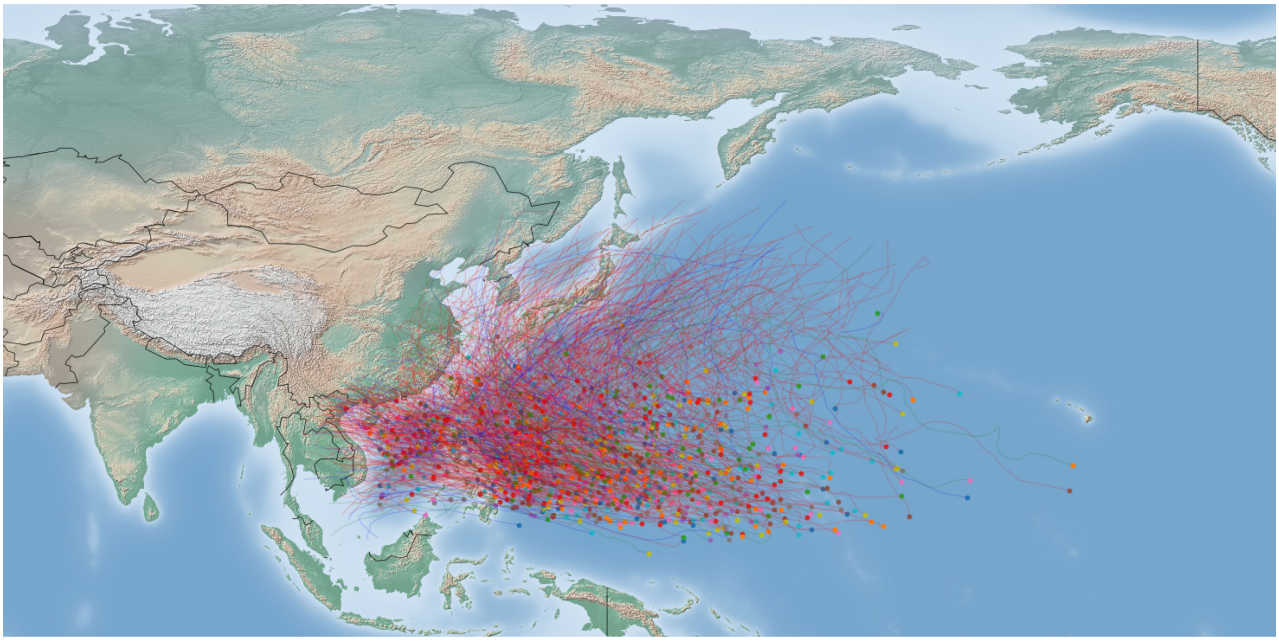


Fig. 7. All TCs track data used in the experiment. The line in red, blue, and green represent the TCs for training, validating, and testing. The scatters in different color represent the first locations of TCs.

TABLE IV
COMPARISON OF THE TCS TRACK FORECASTING RESULTS OF STATISTICAL AND DEEP LEARNING METHODS

| Methods | MDE (km) | | | |
|---|---|---|---|---|
| | 6 h | 12 h | 18 h | 24 h |
| extrapolation | 33.78 | 79.20 | 135.48 | 201.28 |
| CLIPER-BP | 37.53 | 73.31 | 115.13 | 162.62 |
| FFN [13] | 32.90 | - | - | 136.10 |
| BiGRU-attn [14] | - | - | - | 147.38 |
| DBF-Net (Ours) | **31.30** | **58.94** | **87.60** | **119.05** |

The significance of bold entities indicate best values.

TABLE V
COMPARISON OF THE TCS TRACK FORECASTING RESULTS OF NWP METHODS

| Year | | T213/T639 | SHTP | DBF-Net |
|---|---|---|---|---|
| 2014 | #Samples | 411 | 112 | 330 |
| | 24h MDE (km) | 121.8 | **64.9** | 107.4 |
| 2015 | #Samples | 46 | 440 | 671 |
| | 24h MDE (km) | 150.6 | **67.8** | 107.0 |
| 2016 | #Samples | 412 | 194 | 332 |
| | 24h MDE (km) | 114.9 | **88.5** | 118.9 |
| 2017 | #Samples | 301 | 253 | 319 |
| | 24h MDE (km) | 98.7 | **89.1** | 130.6 |
| **AVG** | | 121.5 | **77.6** | 116.0 |
| **Inference time** | | - | - | **2.03s** |

The results of NWP methods is released by [22]–[25].
The significance of bold entities indicate best values.

### E. Comparison With NWP Forecast Methods

We further compare our DBF-Net with the NWP system that commonly used in operational forecasting. The global pattern T213/T639 released by CMA and Shanghai typhoon region pattern (SHTP) released by Shanghai Typhoon Institute of the CMA [22]–[25] are chosen for comparison. Compared with our deep-learning-based method, the NWP methods always need a great number of computation resources and the inference time increases rapidly as input data resolution increases. However, NWP methods still can achieve better forecast accuracy compared with deep learning based methods. As shown in Table V, DBF-Net could achieve the comparable performance compared with global pattern T213/T639, especially in year 2014 and 2015, the 24 h MDE of DBF-Net is much better than T213/T639. However, compared with the region pattern SHTP, the forecast of the DBF-Net still has a certain gap. The great performance of the SHTP may due to the high-resolution multilayer nested grid input data and huge computational resource consumption. In contrast, our proposed DBF-Net achieves relatively high prediction accuracy under the premise of low-resolution input (1° spatio resolution for GPH data) and small computational resource consumption. The inference time of our DBF-Net for multihorizon forecast is only 2.03 s, which is about three to four orders of magnitude faster compared with the NWP methods. Besides, we believe that by using higher resolution data for model training, our DBF-Net could further improve the forecast accuracy, which can be further studied in the future works.

### F. Ablation Study

*Verification of Model Structure:* In order to verify the effectiveness of our proposed DBF-Net architecture, we experimented with different branching structures. As can be seen in

TABLE VI
IMPACT OF DIFFERENT BRANCHES IN DBF-NET

| Architecture | MDE (km) | | | |
|---|---|---|---|---|
| | 6h | 12h | 18h | 24h |
| TC-features-only | 32.87 | 69.08 | 111.08 | 158.11 |
| Pressure-fields-only | 34.35 | 64.07 | 95.83 | 130.79 |
| Pressure-fields-only⋆ | 34.27 | 63.31 | 93.63 | 126.89 |
| DBF-Net | **31.18** | 59.13 | 88.82 | 121.79 |
| DBF-Net⋆ | 31.30 | **58.94** | **87.60** | **119.05** |

⋆ Represents the model with 3D-CNN Decoder in pressure branch to enhance the temporal dynamic changing of GPH data.
The significance of bold entities indicate best values.

Table VI, our proposed DBF-Net with two branches encoder–decoder networks and feature fusion module, which is denoted as "DBF-Net⋆," achieves the best forecast MDE except with the 6 h forecast result in "DBF-Net" that verified the consistency of our method. In addition, the relatively bad performance of LSTM-based and 3D-CNN-based encoder–decoder architecture alone, which is denoted as "TC-features-only" and "'pressure-fields-only," demonstrate that it is hard to obtain good forecast results for single inherent features or meteorological fields input. It makes sense to fuse these two types of data for better forecast accuracy. The results in Table VI also show the effectiveness of the 3D-CNN decoder module for enhancing the temporal dynamic changing of GPH data.

It should be noted that the 3D-CNN decoder module in pressure fields branch only plays a role in model training. Once the DBF-Net is trained, the inference is done by just passing TC features branch and 3D-CNN encoder module. Therefore, the 3D-CNN decoder module does not increase the memory and computational cost of the DBF-Net inference.

*Effectiveness of Multistages Training Method:* As mentioned in Section II-G, we train our DBF-Net in a multistages manner to enhance the ability of feature extraction from different input data of each module in DBF-Net. In order to verify the effectiveness of the multistages training method, we also trained our DBF-Net in an end-to-end manner and the 24 h track forecast accuracy is 128.61 km, which is 9.56 km worse compared with the one with multistages training.

*Forecast Accuracy for TCs of Different Intensity Levels:* We classify TCs according to the intensity level at the initial forecast time. That is tropical depression, tropical storm (TS), severe tropical storm (STS), typhoon (TY), severe typhoon (STY), and super severe typhoon (SuperTY). Fig. 8 reports the track forecast error of various TCs' intensity. It shows that, the forecast error decrease consistently as the intensity of TCs increase. It means that the performance of DBF-Net for forecasting TCs' track with low intensity is relatively poor. The same conclusion is also reported in previous works [13].

## G. Case Study

In this section, we select the following three individual cases of TC events, namely Typhoon Trami 1824, Typhoon Hagibis 1919, and Typhoon Fengshen 1925 (the four digits after the typhoon name represent the typhoon number, for example, the typhoon 1919 represents the nineteenth typhoon of year 2019). According to the forecast results, the validity of the DBF-Net is further verified, and the forecast characteristics of DBF-Net are analyzed.

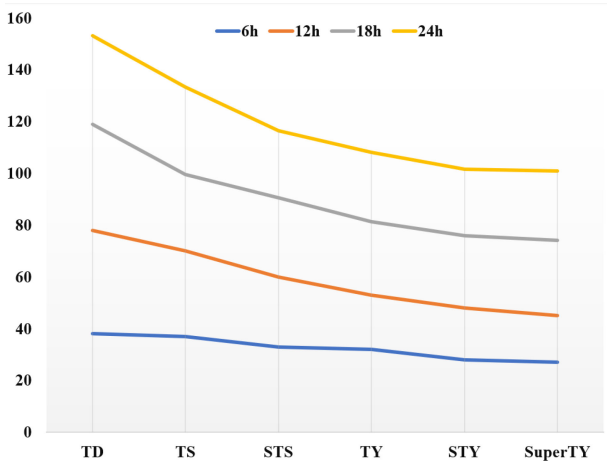



Fig. 8. TCs track forecast result of our proposed DBF-Net of various intensity.

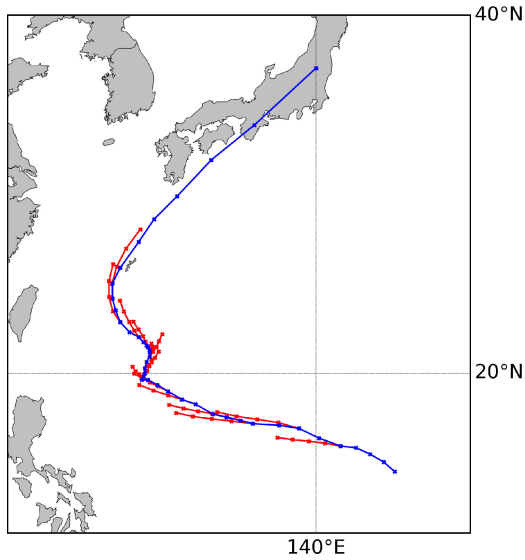



Fig. 9. TCs track forecast results for typhoon Trami 1824. The blue line represents the ground truth track. The red line represents the forecast results.

Figs. 9–11 and Tables VII–IX show the TCs track forecast results for the three cases. In the figures, the blue line represents the ground truth track of TCs and the red line is the forecast results of the proposed DBF-Net. The intersection of the blue and red lines is the location of initial forecast time. The four points extending from the red line represent the forecast path in the next 24 h (6-h interval) from the initial forecast time.

For Typhoon Trami 1824, its path generally shows a trend of first westward and then northward. As shown in Fig. 9, the error between the forecast and the ground truth track is relatively small at the inflection point from west to north. This shows that the model itself has learned the potential features of TCs track movement. For Typhoon Hagibis 1919, there also

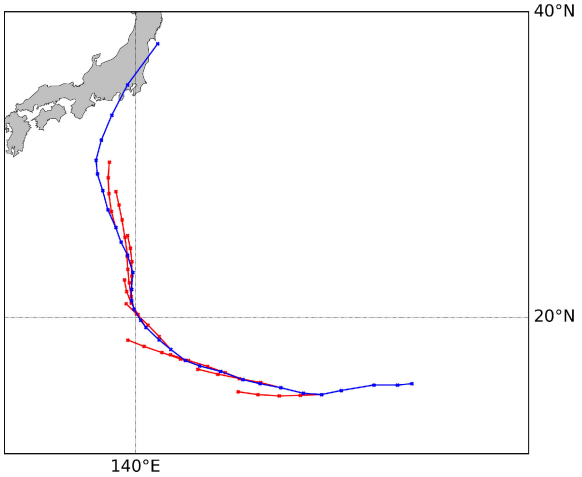



Fig. 10. TCs track forecast results for typhoon Hagibis 1919. The blue line represents the ground truth track. The red line represents the forecast results.

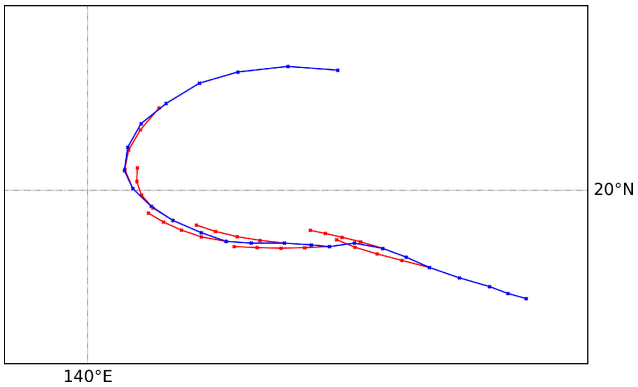



Fig. 11. TCs track forecast results for typhoon Fengshen 1925. The blue line represents the ground truth track. The red line represents the forecast results.

TABLE VII
TRACK FORECAST RESULT OF DBF-NET FOR TYPHOON TRAMI 1824

| Typhoon Trami 1824 | | | | | |
|---|---|---|---|---|---|
| MMDDHH | INT | MDE (km) | | | |
| | | 6h | 12h | 18h | 24h |
| 092118 | TS | 54.56 | 111.16 | 137.78 | 194.56 |
| 092206 | STS | 19.60 | 41.45 | 40.61 | 75.08 |
| 092218 | TY | 63.42 | 103.65 | 136.77 | 147.31 |
| 092306 | STY | 5.89 | 49.85 | 62.80 | 92.46 |
| 092318 | STY | 1.05 | 26.49 | 46.63 | 68.26 |
| 092406 | SuperTY | 11.45 | 21.48 | 11.23 | 66.31 |
| 092418 | SuperTY | 22.59 | 33.89 | 65.07 | 83.58 |
| 092506 | SuperTY | 8.75 | 23.03 | 34.63 | 20.56 |
| 092518 | SuperTY | 24.11 | 27.85 | 52.71 | 62.20 |
| 092606 | STY | 23.19 | 51.10 | 83.84 | 129.74 |
| 092618 | STY | 26.29 | 44.59 | 65.90 | 94.28 |
| 092706 | STY | 13.92 | 28.89 | 62.13 | 86.24 |
| 092718 | STY | 33.67 | 58.95 | 55.01 | 50.70 |
| 092806 | STY | 18.94 | 24.64 | 26.86 | 52.50 |
| 092818 | STY | 3.85 | 23.75 | 91.57 | 108.31 |
| **AVG** | | 23.48 | 41.70 | 58.40 | 82.43 |

INT stands for the intensity level of the TC. AVG stands for the average prediction MDE.

TABLE VIII
TRACK FORECAST RESULT OF DBF-NET FOR TYPHOON HAGIBIS 1919

| Typhoon Hagibis 1919 | | | | | |
|---|---|---|---|---|---|
| MMDDHH | INT | MDE (km) | | | |
| | | 6 h | 12 h | 18 h | 24 h |
| 100606 | STS | 28.26 | 68.36 | 90.68 | 107.11 |
| 100618 | TY | 11.00 | 30.92 | 31.60 | 30.41 |
| 100706 | SuperTY | 38.75 | 62.95 | 26.88 | 41.82 |
| 100718 | SuperTY | 42.27 | 75.44 | 132.27 | 177.07 |
| 100806 | SuperTY | 26.80 | 24.93 | 60.36 | 77.02 |
| 100818 | SuperTY | 51.18 | 54.16 | 82.65 | 93.71 |
| 100906 | SuperTY | 30.06 | 56.08 | 45.70 | 9.55 |
| 100918 | SuperTY | 29.00 | 56.08 | 86.27 | 109.05 |
| 101006 | SuperTY | 47.06 | 75.26 | 90.90 | 99.15 |
| 101018 | STY | 28.45 | 52.75 | 84.87 | 97.22 |
| **AVG** | | 29.43 | 51.41 | 66.84 | 86.77 |

INT stands for the intensity level of the TC. AVG stands for the average prediction MDE.

TABLE IX
TRACK FORECAST RESULT OF DBF-NET FOR TYPHOON FENGSHEN 1925

| Typhoon Fengshen 1925 | | | | | |
|---|---|---|---|---|---|
| MMDDHH | INT | MDE (km) | | | |
| | | 6h | 12h | 18h | 24h |
| 111206 | TS | 34.07 | 50.98 | 26.22 | 63.83 |
| 111218 | TS | 39.65 | 99.83 | 116.08 | 183.11 |
| 111306 | STS | 44.77 | 38.92 | 48.00 | 60.31 |
| 111318 | STS | 59.19 | 75.96 | 90.71 | 152.12 |
| 111406 | STS | 27.19 | 81.96 | 126.19 | 183.04 |
| 111418 | TY | 14.69 | 68.90 | 102.61 | 140.09 |
| 111506 | STY | 9.65 | 17.76 | 38.13 | 50.84 |
| **AVG** | | 28.05 | 62.49 | 92.39 | 137.03 |

INT stands for the intensity level of the TC. AVG stands for the average prediction MDE.

a inflection point at time "100718" (as shown in Fig. 10). Although the forecast result is relatively bad (177.07 km for 24 h forecasting), the DBF-Net also could fix the prediction by bringing the observations from the next time step (24.93 km for 12 h forecasting at time "100806"). This shows that the historical information closest to the initial forecast time is more important. For Typhoon Fengshen 1925, its trajectory presents a 180° turning trend, which is unconventional. As shown in Fig. 11, DBF-Net could correctly predict the turning trend of the TC. However, the forecasting length of the track vector is uniformly smaller than the ground truth and causes the average MDE to be relatively large (see Fig. 11).

In Tables VII–IX, we also report the intensity level (INT) at each time step. By comparing the results in Tables VII–IX, it can be found that the DBF-Net has a relatively lower MDE when the intensity level is larger. Especially for the TCs with the intensity level of "SuperTC," the predicted MDE is comparable to the NWP model. This phenomenon also explains the relatively poor forecast results of Typhoon Fengshen 1925.

## IV. Discussion and Future Work

The proposed DBF-Net provides a relatively efficient way to fuse multimodal spatio-temporal data for TC track forecasting. The extensive experiments further verify the importance of fusing the two types of data for better forecast accuracy. However, as shown in Table V, the forecast results of deep learning-based DBF-Net still have a certain gap compared with the traditional complex NWP methods, although DBF-Net has a greater advantage in terms of computational consumption. Fortunately, recent works on precipitation forecasting demonstrate the potential of deep learning-based methods for weather/climate forecasting [31], [32]. Therefore, there are still many aspects to be studied for deep learning-based TC track forecast methods in the future to improve the forecast accuracy. The following four issues could be considered in the future.

1) Recent deep learning-based precipitation forecasting methods mainly treat the forecast problem as a video frame prediction task [31], [32]. The large-scale image-like data with a high temporal resolution, such as radar map and satellite image, is the primary input of the proposed methods. Although the similar idea also investigates in TCs track forecast area recently [16], the model architecture and scale of the training set are still limited compared with the MetNet [31], and DGMR [32], which can be further studied.
2) The experiment results in Table VI show the relatively bad performance for forecasting only based on TC features, especially for long-term forecast results (24 h). The reason for this may be due to the inadequacy of the LSTM structure. The popular transformer-based model with attention mechanism [4], [5] could be a great choice for capturing the temporal features of sequence data more efficiently.
3) The assumption of regular mesh of the reanalysis 3-D pressure data is adopted in this article for simplicity. And the vanilla convolution operation in 3-D version is used to extract the spatio-temporal features. However, there exist irregular distribution of meteorology field across the different spatial location. In order to solve this problem, the graph convolution [33] and spherical CNNs [34] could be used for further study.
4) Different from the data-driven deep learning-based forecast methods, the NWP methods mainly follow the physics laws by solving the PDEs with few boundary and initial condition samples and achieving outstanding forecast results. These physics laws can be treated as prior information for TCs track forecast, which is still ignored by deep learning-based methods. Recently, many works try to merge the physics information into the deep learning framework for weather forecasting to maintain the advantages of numerical and deep learning-based methods at the same time [35]–[37], so as to improve the performance of deep learning-based TCs track forecasting.

## V. Conclusion

In this article, we study the TC track forecasting for multiple future time steps by proposing a novel deep learning based model, named DBF-Net, to make full use of both inherent features of TCs and reanalysis 3-D pressure fields data and fuse the multimodal features efficiently. DBF-Net contains two branches with encoder-decoder architectures and can be split into three part. The first part is the LSTM-based TC features encoder module that captures the high-level temporal features from historical inherent features of TCs. The second part is the pressure field branch that extracts the spatio-temporal features by learning the temporal dynamic changing of GPH data with a 3D-CNN-based encoder–decoder network. The last part is an LSTM-based decoder module that fuses the multimodal high-level features from different branches and produces the multihorizon prediction.

Experiments on 39 years of historical TCs track dataset in the WNP Ocean show that the 24 h track forecast accuracy reaches 119.05 km, which outperforms previous statistical/deep learning forecast methods. Our DBF-Net also achieves comparable performance compared with the global pattern T213/T639 with orders of magnitude faster. The case study of three different typhoon events further shows the ability of predicting the turning trend of our proposed method.

## References


[1] J. N. Shoosmith, "Numerical analysis," in *Encyclopedia of Physical Science and Technology*, 3rd ed. R. A. Meyers, Ed. New York, NY, USA: Academic, 2003, pp. 39–70.

[2] J. C. Neumann and M. B. Lawrence, "An operational experiment in the statistical-dynamical prediction of tropical cyclone motion," *Monthly Weather Rev.*, vol. 103, no. 8, pp. 665–673, 1975.

[3] Z. Zou *et al.*, "Object detection in 20 years: A survey," 2019, *arXiv:1905.05055*.

[4] V. Ashish *et al.*, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[5] J. Peng *et al.*, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11106–11115.

[6] R. Chen, W. Zhang, and X. Wang, "Machine learning in tropical cyclone forecast modeling: A review," *Atmosphere*, vol. 11, no. 7, 2020, Art. no. 676.

[7] C. M. Kishtawal, M. M. Ali, and S. Jain, "Predicting cyclone tracks in the North Indian ocean: An artificial neural network approach," *Geophys. Res. Lett.*, vol. 34, no. 4, 2007.

[8] W. Zhang, Y. Wang, and W. Fu, "Back propogation(BP)-neural network for tropical cyclone track forecast," in *Proc. 19th Int. Conf. Geoinformat.*, 2011, pp. 1–4.

[9] M. M. Kordmahalleh, M. G. Sefidmazgi, and A. Homaifar, "A sparse recurrent neural network for trajectory prediction of atlantic hurricanes," in *Proc. Genet. Evol. Comput. Conf.*, 2016, pp. 957–964.

[10] A. Sheila, B. Jonathan, P. Adrian, and G. Sam, "Predicting hurricane trajectories using a recurrent neural network," in *Proc. 33rd AAAI Conf. Artif. Intell., 31st Innov. Appl. Artif. Intell. Conf., 9th AAAI Symp. Educ. Adv. Artif. Intell.*, 2019, pp. 468–475.

[11] S. Gao, P. Zhao, B. Pan, Y. Li, and M. Zhou, "A nowcasting model for the prediction of Typhoon tracks based on a long short term memory neural network," *Acta Oceanologica Sinica*, vol. 37, no. 5, pp. 12–16, 2018.

[12] S. Kim *et al.*, "Deep-Hurricane-tracker: Tracking and forecasting extreme climate events," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2019, pp. 1761–1769.

[13] S. Giffard-Roisin, M. Yang, G. Charpiat, C. Kumler-Bonfanti, B. Kégl, and C. Monteleoni, "Tropical cyclone track forecasting using fused deep learning from aligned reanalysis data," *Frontiers Big Data*, p. 1, 2020.

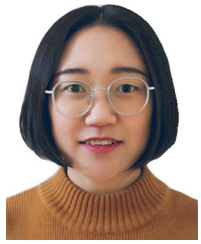
[14] F. Meng *et al.*, "A novel deep learning model by Bigru with attention mechanism for tropical cyclone track prediction in the Northwest Pacific," *J. Appl. Meteorol. Climatol.*, vol. 61, no. 1, pp. 3–12, 2022.

[15] M. Rüttgers, S. Lee, S. Jeon, and D. You, "Prediction of a Typhoon track using a generative adversarial network and satellite images," *Sci. Rep.*, vol. 9, pp. 1–15, 2019.

[16] Y. Wu, X. Geng, Z. Liu, and Z. Shi, "Tropical cyclone forecast using multitask deep learning framework," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6503505.
[17] M. Mudigonda *et al.*, "Segmenting and tracking extreme climate events using neural networks," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017.
[18] M. Rüttgers, S. Lee, and D. You, "Prediction of Typhoon tracks using a generative adversarial network with observational and meteorological data," 2018, *arXiv:1812.01943*.
[19] B. Lim, Ö. Sercan, N. Arık Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *Int. J. Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.
[20] S Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[21] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
[22] Q. Cao, G. Chen, and L. Bai, "Verification on forecast of tropical cyclones over western north Pacific in 2014 (in Chinese)," *Meteorological*, vol. 41, no. 12, pp. 1554–1561, 2015.
[23] L. Bai, G. Chen, and R. Wan, "Verification on forecast of tropical cyclones over western north Pacific in 2015 (in Chinese)," *Meteorological*, vol. 43, no. 4, pp. 501–507, 2017.
[24] L. Bai, G. Chen, X. Zhang, and R. Wan, "Verification on forecast of tropical cyclones over western north Pacific in 2016 (in Chinese)," *Meteorological*, vol. 44, no. 4, pp. 582–589, 2018.
[25] L. Bai, G. Chen, X. Zhang, and R. Wan, "Verification on forecast of tropical cyclones over western north Pacific in 2017 (in Chinese)," *Meteorological*, vol. 45, no. 4, pp. 577–586, 2019.
[26] K. Andrej, J. Justin, and F. F. Li, "Visualizing and understanding recurrent networks," 2015, *arXiv:1506.02078*.
[27] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.
[28] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, 2014, pp. 1724–1734.
[29] M. Ying *et al.*, "An overview of the China Meteorological Administration tropical cyclone database," *J. Atmospheric Ocean. Technol.*, vol. 31, no. 2, pp. 287–301, 2014.
[30] S. Saha *et al.*, "The NCEP climate forecast system reanalysis," *Bull. Amer. Meteorological Soc.*, vol. 91, no. 8, pp. 1015–1058, 2010.
[31] C. K. Sønderby *et al.*, "METNet: A neural weather model for precipitation forecasting," 2020, *arXiv:2003.12140*.
[32] S. Ravuri *et al.*, "Skilful precipitation nowcasting using deep generative models of radar," *Nature*, vol. 597, pp. 672–677, 2021.
[33] H. Lin, Z. Gao, Y. Xu, L. Wu, L. Li, and S. Z. Li, "Conditional local convolution for spatio-temporal meteorological forecasting," 2021.
[34] C. M. Jiang, J. Huang, K. Kashinath, P. Marcus, and M. Niessner, "Spherical CNNs on unstructured grids," 2019, *arXiv:1901.02039*.
[35] K. Kashinath, M. Mustafa, A. Albert, J. L. Wu, and B. Prabhat, "Physics-informed machine learning: Case studies for weather and climate modelling," *Philos. Trans. Roy. Soc. A Math. Phys. Eng. Sci.*, vol. 379, no. 2194, 2021, Art. no. 20200093.
[36] L. Espeholt *et al.*, "Skillful twelve hour precipitation forecasts using large context neural networks," 2021, *arXiv:2111.07470*.
[37] R. Wang, K. Kashinath, M. Mustafa, A. Albert, and R. Yu, "Towards physics-informed deep learning for turbulent flow prediction," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, pp. 1457–1466, 2020.

**Zili Liu** received the B.S. and M.S. degrees in statistics from the School of Science, China University of Mining and Technology, Beijing, China, in 2017 and 2021, respectively. He is currently working toward the Ph.D. degree in pattern recognition and intelligent systems with the Image Processing Center, School of Astronautics, Beihang University, Beijing, China.

His research interests include machine learning, pattern recognition, and efficient deep learning.



**Kun Hao** received the B.S. degree in detection guidance and control and M.S. degree in aerospace engineering from the School of Astronautics, Beihang University, China, in 2018 and 2021, respectively.

Her research interests include image processing and deep learning.

**Xiaoyi Geng** received the B.S. degree in detection guidance and control and the M.S. degree in aerospace engineering from the School of Astronautics, Beihang University, China, in 2019 and 2022, respectively.

Her research interests include image processing and deep learning.

**Zhengxia Zou** received the B.S. and Ph.D degrees in pattern recognition and intelligent systems from Beihang University, Beijing, China, in 2013 and 2018, respectively.

He is currently an Associate Professor with the School of Astronautics, Beihang University. During 2018–2021, he was a Postdoctoral Research Fellow with the University of Michigan, Ann Arbor, Ann Arbor, MI, USA. His research interests include computer vision and related problems in remote sensing and autonomous driving. He has authored or coauthored more than 20 peer-reviewed papers in top-tier journals and conferences, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, Conference on Computer Vision and Pattern Recognition, International Conference on Computer Vision, Association for the Advancement of Artificial Intelligence. His research has been featured in more than 30 global tech media outlets and adopted by multiple application platforms with more than 50 million users worldwide.

**Zhenwei Shi** (Member, IEEE) received the Ph.D. degree in mathematics from the Dalian University of Technology, Dalian, China, in 2005.

He was a Postdoctoral Researcher with the Department of Automation, Tsinghua University, Beijing, China, from 2005 to 2007. He was a Visiting Scholar with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA, from 2013 to 2014. He is currently a Professor and the Dean of the Image Processing Center, School of Astronautics, Beihang University, Beijing, China. He has authored or coauthored more than 100 scientific papers in refereed journals and proceedings, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, and IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. His current research interests include remote sensing image processing and analysis, computer vision, pattern recognition, and machine learning.

Dr. Shi is currently an Associate Editor for the Infrared Physics and Technology.