WeatherGNN: Exploiting Meteo- and Spatial-Dependencies for Local Numerical Weather Prediction Bias-Correction

Binqing Wu[1,2] , **Weiqi Chen**[1] , **Wengwei Wang**[1] , **Bingqing Peng**[1] , **Liang Sun**[1] and **Ling Chen**[2]
[1] DAMO Academy, Alibaba Group
[2] College of Computer Science and Technology, Zhejiang University
{binqingwu, lingchen}@cs.zju.edu.cn,
{jarvus.cwq, duoluo.www, pengbingqing.pbq, liang.sun}@alibaba-inc.com

Abstract

Due to insufficient local area information, numerical weather prediction (NWP) may yield biases for specific areas. Previous studies correct biases mainly by employing handcrafted features or applying data-driven methods intuitively, overlooking the complicated dependencies between weather factors and between areas. To address this issue, we propose WeatherGNN, a local NWP bias-correction method that utilizes Graph Neural Networks (GNNs) to exploit meteorological dependencies and spatial dependencies under the guidance of domain knowledge. Specifically, we introduce a factor GNN to capture area-specific meteorological dependencies adaptively based on spatial heterogeneity and a fast hierarchical GNN to capture dynamic spatial dependencies efficiently guided by Tobler's first and second laws of geography. Our experimental results on two real-world datasets demonstrate that WeatherGNN achieves the state-of-the-art performance, outperforming the best baseline with an average of 4.75 % on RMSE.

1 Introduction

Numerical weather prediction (NWP) has become the widely accepted and effective method for weather forecasting [Bauer *et al.*, 2015], developing from solving mathematical equations under physical laws [Moran and Moran, 2009] to incorporating deep learning methods [Bi *et al.*, 2023; Chen *et al.*, 2023b; Nguyen *et al.*, 2023]. Despite its advancements, due to insufficient local area information [Gneiting and Raftery, 2005; Yoshikane and Yoshimura, 2022], NWP may be still biased for specific areas. This kind of deviation leads to significant discrepancies in downstream applications, e.g., wind power forecasting [Han *et al.*, 2022].

Local NWP bias-correction aims to correct biases of NWP data for specific areas, given additional local area information, e.g., weather factor observations and terrain [Zhang *et al.*, 2023a]. Early local NWP bias-correction methods usually apply statistical rules and shallow machine learning [Delle Monache *et al.*, 2006; Cui *et al.*, 2012; Durai and Bhradwaj, 2014], which cannot model the complex non-linearity within NWP. Recently, some deep learning methods have achieved impressive success owing to their strong representative abilities. They mainly rely on Recurrent Neural Networks (RNNs) [Li *et al.*, 2022; Yang *et al.*, 2022], Convolutional Neural Networks (CNNs) [Han *et al.*, 2021], or their hybrid methods [Han *et al.*, 2022]. Despite the promising results of existing methods, we argue that two domain-specific problems are still overlooked.

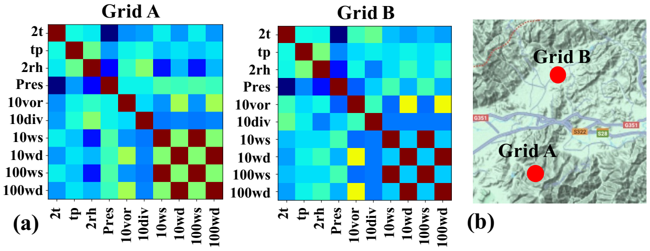

Figure 1: (a) Pearson Correlation Matrices between Weather Factors at Different Grids. (b) Locations and Terrains of Two Grids. Grid A is on an uphill, while B is in a valley. The distance between Grid A and B is around 18 kilometers.

First, these methods mainly correct weather factors independently, rarely considering the complicated dependencies between them. In reality, weather factors interact with each other and these dependencies are strongly correlated with heterogeneous geographical characteristics, e.g., terrain. Fig. 1 illustrates an example, where each grid represents a specific area. Grid A, positioned uphill, and Grid B, located in a valley, exhibit distinct correlation patterns even if they are close.

Second, previous methods often neglect or only concentrate on grid-based dependencies between areas, failing to capture complicated dependencies between areas. Due to geographical impacts and atmospheric motions, the weather conditions of one area have a grid-agnostic and dynamic influence on those of other areas. Fig. 2 illustrates an example of 100m wind speed. Over a half-month duration, the similarity of wind speed between Grid P and other areas closely aligns with the terrain similarity, as depicted in Fig. 2(b), showcasing grid-agnostic dependencies. Meanwhile, the similarity of wind speed between Grid P and Grid Q, computed within a six-hour time range of that duration, undergoes temporal evolution, as revealed in Fig. 2(c).

To this end, we propose WeatherGNN, a GNN-based method that exploits meteorological dependencies and spatial

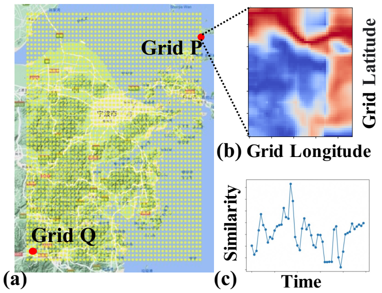

Figure 2: (a) Grid Distribution and Terrain of Ningbo Dataset. (b) DTW Similarity of Half-month 100m Wind Speed between Grid P and Other Grids. (c) DTW Similarity of 100m Wind Speed with a Six-hour Time Range between Grid P and Grid Q.

dependencies for local NWP bias-correction. WeatherGNN adopts two GNNs to capture area-specific meteorological dependencies adaptively and dynamic spatial dependencies efficiently, guided by geography principles, i.e., spatial heterogeneity and Tobler's first and second laws. To the best of our knowledge, WeatherGNN is the first work that incorporates GNNs and domain knowledge for this task. The main contributions are as follows:

- Introduce a factor GNN that combines factor embeddings and geographical embeddings derived from heterogeneous geographical characteristics to construct factor graphs for each area, which can learn area-specific meteorological dependencies adaptively.
- Introduce a fast hierarchical GNN that constructs a static hierarchy based on pre-defined geographical and meteorological distances and adjusts the hierarchy according to NWP data, which can capture dynamic spatial dependencies. Notably, we introduce a fast hierarchical message passing module, inspired by Tobler's laws, that exchanges fine-grained messages among grids with stronger dependencies and coarse-grained messages through repeated operations among grids with weaker dependencies, which can achieve linear complexity concerning the number of grids.
- Experimental results on two real-world datasets demonstrate the superiority of WeatherGNN, outperforming the best baselines with an average of 4.75 % on RMSE.

2 Related Work

Local NWP Bias-correction. Early local bias-correction methods evolve from statistical methods, e.g., correcting maximum and minimum values [Durai and Bhradwaj, 2014] or the variance and average [Zhang *et al.*, 2019], to shallow machine learning methods, e.g., applying Kalman filter [Delle Monache *et al.*, 2006] and deep belief networks [Hu *et al.*, 2021], which cannot capture non-linear dependencies within NWP. Recently, some deep learning methods have been applied due to their strong representative abilities. They mainly rely on RNNs [Li *et al.*, 2022; Yang *et al.*, 2022], CNNs [Han *et al.*, 2021], or their hybrid methods [Han *et al.*, 2022; Zhang *et al.*, 2023a]. Some advanced methods have also been applied to local NWP bias-correction. For example, Dl-Corrector-Remapper [Ge *et al.*, 2022] uses Adaptive Fourier Neural Operators (AFNOs) to learn continuous status of the atmosphere and correct four weather factors. CM2Mc-LPJmL [Hess *et al.*, 2022] utilizes Generative Adversarial Networks (GANs) to improve distributions of the precipitation output. However, since they mainly correct one or a few weather factors independently, they rarely consider complicated dependencies between weather factors. In addition, existing methods barely consider or only capture gird-based dependencies between areas via CNNs, which cannot capture gird-agnostic dependencies between areas.

State-of-the-art Deep Learning Methods for Weather Applications. SOTA methods have shown promising results in many weather applications. As discussed above, for local NWP bias-correction, RNNs and CNNs have achieved great improvement. For precipitation prediction, CNNs, GANs, and vision transformers show impressive ability [Ravuri *et al.*, 2021; Zhang *et al.*, 2023b; Gao *et al.*, 2022]. For weather downscaling, CNNs [Hu *et al.*, 2019] and Physics-informed neural networks [Esmaeilzadeh *et al.*, 2020] have garnered advancements. For medium-range global weather forecasting, methods ranging from AFNOs [Pathak *et al.*, 2022] to hierarchical transformers [Chen *et al.*, 2023a; Chen *et al.*, 2023b; Nguyen *et al.*, 2023] have demonstrated significant success [Rasp *et al.*, 2023]. Recently, GraphCast [Lam *et al.*, 2023] designs a multi-scale unweighted mesh graph for global forecasts, which exhibits the strong potential of GNNs to model atmospheric dynamics. However, these methods only focus on simple spatial dependencies and barely consider dependencies between weather factors explicitly.

Graph Neural Networks. GNN is a general framework for constructing neural networks on graph-structured data. The most common form of a GNN is the message-passing paradigm [Wu *et al.*, 2020; Zhou *et al.*, 2020]. During each message-passing layer, the representations of a node are updated according to messages aggregated from the graph neighborhood of that node. The update and aggregate operators are implemented by neural networks. Due to the strong representative ability on graphs, GNNs have achieved impressive success in many real-world applications [Wu *et al.*, 2022], ranging from graph classification [Ma *et al.*, 2023], traffic forecasting [Jiang *et al.*, 2023], and air quality estimation [Chen *et al.*, 2023c]. However, GNNs face the challenge of designing graphs for specific applications and executing efficient message passing across graphs. Exploring methods to harness the advantages of GNNs and customize them for local NWP bias-correction is still an ongoing challenge.

3 WeatherGNN

We formalize local NWP bias-correction and describe how to model area-specific meteorological dependencies and dynamic spatial dependencies using WeatherGNN.

3.1 Problem Definition

The goal of local NWP bias-correction is to correct biases of NWP data for specific areas, given additional local area information. Formally, NWP data for areas are represented as a

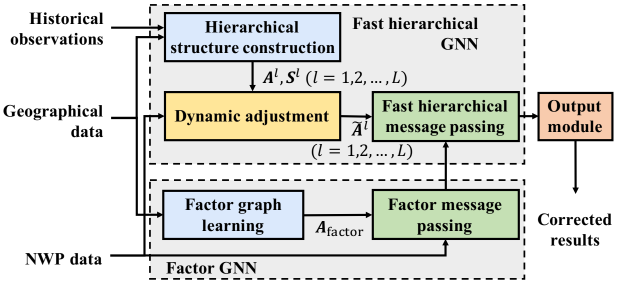

Figure 3: An Overview of WeatherGNN.

sequence $[\mathcal{X}_i]_{i=1}^{T} \in \mathbb{R}^{N\times F_{\rm e}\times T}$, where N denotes the number of grids [1]. $F_{\rm e}$ denotes the number of weather factors, e.g., temperature, humidity, and wind speed, and T is the length of the sequence. Similarly, $[\mathcal{Y}_i]_{i=1}^{T} \in \mathbb{R}^{N\times F_{\rm e}\times T}$ represents the corresponding weather factor observations. Moreover, each grid has associated geographical information denoted as $\mathcal{Z} \in \mathbb{R}^{N\times F_{\rm g}}$, where $F_{\rm g}$ is the dimension of geographic features, e.g., longitude, latitude, and altitude. The local NWP bias-correction is formulated as learning a function $f(\cdot)$ that maps NWP data to local weather factor observations incorporating geographical information:

$$\begin{aligned} &\{[\mathcal{X}_i]_{i\in\Omega(t)};\mathcal{Z}\} \xrightarrow{f(\cdot)} \mathcal{Y}_t, \\ &\Omega(t)=\{t-\tau,\cdots,t-1,t,t+1,\cdots,t+\tau\}, \end{aligned} \tag{1}$$

where $\Omega(t)$ denotes a temporal window around time step t with length $T = 2\tau + 1$, as we consider a period of time before and after t when correcting $\mathcal{X}_t$. For simplicity, in the remainder of the paper, we denote $[\mathcal{X}_i]_{i\in\Omega(t)}$ and $\mathcal{Y}_t$ by $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively.

3.2 Model Overview

As illustrated in Fig. 3, WeatherGNN adopts a two-branch architecture. A factor GNN and a fast hierarchical GNN are designed to capture area-specific meteorological dependencies and dynamic spatial dependencies, respectively. Subsequently, an output module is applied to yield corrected results.

3.3 Learning Area-specific Meteorological Dependencies via Factor GNN

According to ***Spatial heterogeneity*** [Anselin, 2013], which states that geographic variables exhibit uncontrolled variance, meteorological dependencies, significantly influenced by geographic variables, are expected to vary based on heterogeneous geographical characteristics consequently. However, defining meteorological dependencies for each area is challenging due to issues of applicability and non-linearity. To address this problem, we introduce a factor GNN that constructs a factor graph for each area based on the geographical characteristics of each area in a data-driven way.

Factor graph learning. The factor graph learning (FGL) module first randomly initializes a learnable factor embedding dictionary $\boldsymbol{E}_{\rm f} \in \mathbb{R}^{F_{\rm e}\times d_{\rm e}}$ for weather factors, where each row corresponds to the embedding vector for each factor, and $d_{\rm e}$ is the dimension of factor embedding. This embedding is optimized during model training, aiming to capture the latent features associated with each factor. Then, we take the geographical information of each grid $\mathcal{Z}$ as input and encode it with an MLP to obtain the geographical embedding matrix $\boldsymbol{E}_{\rm g} = \mathrm{MLP}(\mathcal{Z}) \in \mathbb{R}^{N\times d_{\rm e}}$, where N is the number of grids. $\boldsymbol{E}_{\rm g}$ represents geographical characteristics of each grid. Given the factor embedding shared by all grids and the geographical embedding of each grid, we can infer dependencies between weather factors for each grid by:

$$\begin{aligned} \boldsymbol{E}_i &= \boldsymbol{E}_{\rm f} + \boldsymbol{E}_{{\rm g},i}, \\ \boldsymbol{A}_{{\rm f},i} &= \mathrm{Softmax}(\boldsymbol{E}_i\boldsymbol{E}_i^{\top}), \\ \boldsymbol{A}_{\rm f} &= \{\boldsymbol{A}_{{\rm f},i}\}_{i=1}^{N}, \end{aligned} \tag{2}$$

where $\boldsymbol{E}_i \in \mathbb{R}^{F_{\rm e}\times d_{\rm e}}$ is the grid-specific factor embedding of grid i considering geographical characteristics. $\boldsymbol{A}_{{\rm f},i} \in \mathbb{R}^{F_{\rm e}\times F_{\rm e}}$ represents the adjacency matrix of the factor graph of grid i. The FGL module provides structured representations of meteorological dependencies for each grid/area, which can facilitate an understanding of how these weather factors interact, taking into account heterogeneous geographical characteristics.

Factor Message Passing. Enhanced by FGL, the factor message passing for a single layer can be formulated as:

$$\boldsymbol{H}_i = (\boldsymbol{I}_{\rm f} + \boldsymbol{A}_{{\rm f},i})\boldsymbol{X}_i\boldsymbol{W}_{{\rm f},i} + \boldsymbol{b}_{{\rm f},i} \tag{3}$$

where $\boldsymbol{X}_i \in \mathbb{R}^{F_{\rm e}\times T}$ is the NWP data of grid i, and $\boldsymbol{H}_i \in \mathbb{R}^{F_{\rm e}\times d}$ is the corresponding output considering meteorological dependencies. $\boldsymbol{I}_{\rm f}$ is the identity matrix. $\boldsymbol{W}_{{\rm f},i} \in \mathbb{R}^{F_{\rm e}\times T\times d}$ and $\boldsymbol{b}_{{\rm f},i} \in \mathbb{R}^{F_{\rm e}\times d}$ denote the learnable weights and bias, respectively. They can be generated by two smaller parameter matrices and grid-specific factor embeddings, i.e., $\boldsymbol{W}_{{\rm f},i} = \boldsymbol{E}_i\boldsymbol{W}_{\rm w}$ and $\boldsymbol{b}_{{\rm f},i} = \boldsymbol{E}_i\boldsymbol{W}_{\rm b}$. Note that $\boldsymbol{W}_{\rm w} \in \mathbb{R}^{d_{\rm e}\times T\times d}$ and $\boldsymbol{W}_{\rm b} \in \mathbb{R}^{d_{\rm e}\times d}$ are shared by all grids. $\boldsymbol{W}_{{\rm f},i}$ and $\boldsymbol{b}_{{\rm f},i}$ can be interpreted as capturing grid-specific patterns guided by a set of shared patterns discovered from all grids. This factor GNN is performed for each grid in parallel, obtaining the final output $\boldsymbol{H} = \{\boldsymbol{H}_i\}_{i=1}^{N} \in \mathbb{R}^{N\times F_{\rm e}\times d}$. According to Lemma 1, the factor GNN has an $O(N)$ complexity.

3.4 Learning Dynamic Spatial Dependencies via Fast Hierarchical GNN

Due to the complex terrains and varying weather conditions, capturing dependencies between areas is challenging. To address this issue, we propose a Fast Hierarchical GNN (FHGNN) that can effectively capture dynamic spatial dependencies by combining geographical distances, historical weather similarity, and NWP data. Specifically, we first calculate the initial spatial proximity of grids by considering geographical distances and historical weather similarity. Then, we construct the hierarchical structure through an iterative process of clustering grids into multi-level clusters (supergrids). In this structure, grids within the same cluster at lower levels exhibit stronger dependencies and are expected to interact more closely with each other. Following this, we optimize spatial proximity by leveraging an attention matrix

[1] Areas tend to appear as a rectangle containing $H \times W$ grids, and here we flatten the shape with $N = H \times W$.

derived from NWP data, considering current weather conditions. Subsequently, we introduce a Fast Hierarchical Message Passing (FHMP) module to capture dynamic spatial dependencies efficiently. This module facilitates fine-grained interactions between grids at a lower level and coarse-grained interactions at a higher level. Notably, FHGNN demonstrates a linear complexity concerning the number of grids, in contrast to the quadratic complexity of a vanilla GNN.

Constructing static hierarchical structure based on predefined geographical and meteorological distances. According to ***Tobler's first law of geography*** [Miller, 2004], which states that everything is related to everything else, but near things are more related than distant things, we first determine the neighboring grids for each grid. While the Euclidean distance is an intuitive measure of spatial proximity, it falls short of capturing the diverse surface features.

Since variations in altitude significantly influence hydrology, vegetation, and other geographical phenomena, we consider altitude as a crucial factor reflecting geographical features and the Earth's surface shape, in addition to latitude and longitude. We compute the 3D geographical distance matrix $\boldsymbol{D}$ using latitude, longitude, and altitude data obtained from geographical information $\mathcal{Z}$. In addition, as illustrated in Fig. 2, historical weather similarity can also indicate terrain similarity. Thus, we calculate the meteorological distance matrix $\boldsymbol{W}$ by employing DTW [Müller, 2007] to measure the distance between weather sequences for each grid. The two distance matrices are calculated as follows:

$$\boldsymbol{D}_{ij} = \sqrt{\lambda_{\text{lat}} d^2_{\text{lat},ij} + \lambda_{\text{lon}} d^2_{\text{lon},ij} + \lambda_{\text{alt}} d^2_{\text{alt},ij}}, \tag{4}$$

$$\boldsymbol{W}^f_{ij} = \text{DTW}(\mathbf{Y}^f_i, \mathbf{Y}^f_j), \boldsymbol{W}_{ij} = \frac{1}{F_{\text{e}}} \Sigma^{F_{\text{e}}}_{f=1} \boldsymbol{W}^f_{ij}, \tag{5}$$

where $d_{\text{lat},ij}$, $d_{\text{lon},ij}$, $d_{\text{alt},ij}$, are the latitude, longitude, and altitude distance between grid i and j, respectively, with $\lambda_{(\cdot)}$ being the corresponding coefficients to adjust the importance of these distances for calculating the geographical distance. $\mathbf{Y}^f_i$ and $\mathbf{Y}^f_j$ are the observation time series of the f-th weather factor of grid i and j in the whole training data, respectively. $\boldsymbol{W}^f_{ij}$ and $\boldsymbol{W}_{ij}$ are the DTW distance matrix of grid i and j for weather factor f and for all weather factors, respectively.

Gaussian kernels are then employed to calculate two initial adjacency matrices, and these matrices are combined to form the spatial adjacency matrix. The process is formulated as:

$$\begin{aligned} &\boldsymbol{A}_{\text{geo}} = \exp(-\boldsymbol{D}^2/\sigma^2_{\text{geo}}), \quad \boldsymbol{A}_{\text{met}} = \exp(-\boldsymbol{W}^2/\sigma^2_{\text{met}}), \\ &\boldsymbol{A}_{\text{spatial}} = \boldsymbol{A}_{\text{geo}} + \boldsymbol{A}_{\text{met}}, \end{aligned} \tag{6}$$

where $\sigma_{(\cdot)}$ are hyperparameters to control the scale of the corresponding Gaussian kernels, which ensures both matrices have a similar range of values.

Since grids have different neighbors at different spatial scales, we adopt a clustering algorithm, e.g., K-means, to cluster grids into multi-level super-grids based on $\boldsymbol{A}_{\text{spatial}}$. This process forms hierarchical graphs, offering insights into the neighbors of each grid at different spatial scales. The clustering procedure is iterated L times. At the l-th iteration, grids at level l are clustered into the next coarsened super-grids at level $l+1$. The assignment matrix from this clustering, denoted as $\boldsymbol{S}^l \in \{0,1\}^{N^l \times N^{l+1}}$, signifies the assignment of each grid at level l to a super-grid at level $l+1$, with each row containing a single 1 and the rest as 0s. Here, N^l represents the number of grids at level l. In addition, we compute the spatial adjacency matrix $\boldsymbol{A}^l$ for each level. The construction procedure of the hierarchical structure can be formulated as:

$$\begin{aligned} &\boldsymbol{S}^l \leftarrow \text{K-means}(\boldsymbol{A}^l), \\ &\boldsymbol{A}^l = (\boldsymbol{S}^{l-1})^\top \boldsymbol{A}^{l-1} \boldsymbol{S}^{l-1}, \quad \boldsymbol{A}^0 = \boldsymbol{A}_{\text{spatial}}, \end{aligned} \tag{7}$$

where the spatial adjacency matrix $\boldsymbol{A}^l \in \mathbb{R}^{N^l \times N^l}$ ($l = 1, 2, \cdots, L$) signifies a coarsened spatial graph that captures the connectivity strength between super-grids at level l.

Moreover, we introduce the mask matrix $\boldsymbol{M}^l$ to make $\boldsymbol{A}^l$ sparse. The (i,j)-th entry $\boldsymbol{M}^l_{ij}$ is set to 1 if grids i and j belong to the same cluster at level $l+1$, and 0 otherwise. This assignment is based on the observation that grids within the same cluster exhibit stronger spatial dependencies, making them more desirable to retain. Consequently, $\boldsymbol{M}^l$ reflects the most crucial neighboring grids at level l.

Dynamic adjustment of hierarchical structure based on NWP data. Spatial dependencies are further influenced by dynamic weather conditions. For example, on a windy day, a grid has stronger dependencies with its upwind neighboring grids compared to a windless day. Thus, we encode the input NWP data $\boldsymbol{X}$ and utilize the attention matrix to adaptively adjust the spatial adjacency matrices. Inspired by attention mechanisms [Vaswani *et al.*, 2017], this dynamic adjustment is formulated as [2]:

$$\begin{aligned} &\boldsymbol{H}^0_{\text{Q}} = \boldsymbol{X}\boldsymbol{W}_{\text{Q}}, \quad \boldsymbol{H}^0_{\text{K}} = \boldsymbol{X}\boldsymbol{W}_{\text{K}}, \\ &\boldsymbol{H}^l_{\text{Q}} = (\boldsymbol{S}^{l-1})^\top \boldsymbol{H}^{l-1}_{\text{Q}}, \quad \boldsymbol{H}^l_{\text{K}} = (\boldsymbol{S}^{l-1})^\top \boldsymbol{H}^{l-1}_{\text{K}}, \end{aligned} \tag{8}$$

$$\tilde{\boldsymbol{A}}^l = \boldsymbol{M}^l \odot \text{Softmax}(\boldsymbol{H}^l_{\text{Q}} {\boldsymbol{H}^l_{\text{K}}}^\top / \sqrt{d} \odot \boldsymbol{A}^l), \tag{9}$$

where $\boldsymbol{H}^0_{\text{Q}}, \boldsymbol{H}^0_{\text{K}} \in \mathbb{R}^{N \times d}$ are calculated by linear projections of input NWP data $\boldsymbol{X} \in \mathbb{R}^{N \times F_{\text{e}} \times T}$ with parameters $\boldsymbol{W}_{\text{Q}}, \boldsymbol{W}_{\text{K}} \in \mathbb{R}^{F_{\text{e}} \times T \times d}$, respectively. These representations are iteratively aggregated according to the cluster assignment matrix, generating representations $\boldsymbol{H}^l_{\text{Q}}, \boldsymbol{H}^l_{\text{K}} \in \mathbb{R}^{N^l \times d}$ for super-grids at each level. $\odot$ denotes element-wise product. Different from existing attention mechanisms, the mask $\boldsymbol{M}^l$ is used to make the spatial adjacency matrix sparse, where only interactions of grids belonging to the same cluster are reserved. Note that this dynamic adjustment only modifies the spatial adjacency matrices at each level and does not change the assignment matrices between levels.

Fast hierarchical message passing. According to ***Tobler'second law of geography*** [Miller, 2004], which states that the phenomenon external to a geographic area of interest affects what goes on inside, we correct NWP biases of

[2] We introduce the aggregation and sparsification procedure in a dense matrix form for ease of understanding, while in our implementation, the cluster assignment, mask, and spatial adjacency are represented by a sparse matrix form.

one area by considering the impact from other areas. However, given that the number of grids in local NWP bias-correction is typically larger than in other common spatial-temporal applications, e.g., traffic forecasting, applying the Vallinia message-passing module with quadratic complexity on the constructed dynamic graphs poses an efficiency challenge. Therefore, we introduce the Fast Hierarchical Message Passing (FHMP) module to capture dynamic spatial dependencies efficiently. Inspired by Tobler's first and second laws of geography, the main idea of FHMP is to conduct fine-grained message passing among low-level grids which have stronger relationships but coarse-grained message passing at high levels which have weaker relationships. The process can be formulated as:

$$\begin{aligned}
\text{Message}: \hat{\boldsymbol{H}}^{l} &= \boldsymbol{H}^{l}\boldsymbol{W}^{l}, \boldsymbol{H}^{l} = (\boldsymbol{S}^{l-1})^{\top}\boldsymbol{H}^{l-1},\\
\text{Aggregation}: \tilde{\boldsymbol{H}}^{l}_{v^l} &= \frac{1}{|\mathcal{N}_{v^l}|}\sum_{u^l}\tilde{\boldsymbol{A}}^{l}_{u^l v^l}\hat{\boldsymbol{H}}^{l}_{u^l},\\
&\forall v^l \in [0,1,2,\cdots N^l-1], \mathcal{N}_{v^l} = \{u^l|\tilde{A}^{l}_{u^l v^l} \neq 0\},\\
\text{Duplication}: \boldsymbol{H}' &= \tilde{\boldsymbol{H}}^{0} + \sum_{l=1}^{L}\prod_{i=0}^{l-1}\boldsymbol{S}^{i}\tilde{\boldsymbol{H}}^{l},\\
\text{Update}: \boldsymbol{H}'' &= \boldsymbol{H} + \boldsymbol{H}'\boldsymbol{W},
\end{aligned} \tag{10}$$

where $\hat{\boldsymbol{H}}^{l} \in \mathbb{R}^{N^l \times d}$ are calculated by linear projections of $\boldsymbol{H}^l$, representing the message of grids at level l, and $\boldsymbol{H}^0 = \boldsymbol{H}$ is the output of the factor GNN (see Eq. 3). For a specific grid v_l at level l, its neighbors $\mathcal{N}_{v^l}$ are identified by the spatial adjacency matrix $\tilde{\boldsymbol{A}}^{l}$. v_l then aggregates its neighbors' messages according to the adjacency weights in $\tilde{\boldsymbol{A}}^{l}$. The duplication stage is the key design of FHMP module [3]. It duplicates the messages from the super-grids at each level to their containing grids at level 0 based on assignment matrices $[S^i]_{i=0}^{L-1}$, and the grids at level 0 belonging to the same super-grid share the same message from $\hat{\boldsymbol{H}}^{l}$, as illustrated in Fig. 4. With the hierarchical structure[4], all grid-pairs at level 0 can interact. After collecting the messages from all levels, the hidden representations of grids at level 0 are updated based on integrated messages with a residual connection. According to Lemma 1, FHMP in FHGNN have a linear complexity w.r.t. the number of grids. The proof is in Appendix.

Lemma 1. The complexity of factor GNN and fast hierarchical GNN is $O(N)$, where N is the number of grids.

3.5 Bias-Correction Output

Given the output $\boldsymbol{H}'' \in \mathbb{R}^{N \times d}$ of FHGNN, we feed it into an MLP-based decoder to produce corrected weather factors for each grid at the target time step at once. We adopt the L1 loss function to compare the difference between corrected results and ground truth, which can be formulated as: $\mathcal{L}(\hat{\boldsymbol{Y}}, \boldsymbol{Y}) = \sum_{f=1}^{F_e} \alpha_f \|\hat{\boldsymbol{Y}}^{f} - \boldsymbol{Y}^{f}\|_1$, where $\hat{\boldsymbol{Y}} = \mathrm{MLP}\left(\boldsymbol{H}''\right)$, and $\hat{\boldsymbol{Y}}^{f}$ and $\boldsymbol{Y}^{f}$ are the corrected results and the ground truth of the weather factor f. α_f is a hyperparameter representing the corresponding weight of weather factor f.

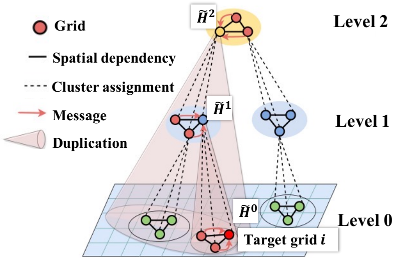

Figure 4: An Overview of Fast Hierarchical Message Passing.

4 Experiments

4.1 Datasets

Since local NWP bias-correction requires high-precision geographical data and direct mapping between NWP and observations of multiple weather factors, current datasets cannot meet such requirements. Thus, we collect two real-world bias-correction datasets: Ningbo and Ningxia, covering two representative terrain types in China. Each grid in both datasets has three types of data, i.e., geographical data, NWP data, and weather factor observations. Geographical data contain latitude, longitude, and altitude. Altitude data are from Digital Elevation Model (DEM) data. In particular, DME data are commonly used in geographic information systems to represent terrain, whose positive and negative values can denote land and ocean, respectively. **Ningbo** dataset depicts a coastline area with latitude range $28.85°\text{N}-30.56°\text{N}$ and longitude range $120.91°\text{E}-122.29°\text{E}$. There are 58×47 grids with a grid size of 0.03 degree in latitude and longitude. Both NWP and observational weather data have hourly records including 10 weather factors from 1/Jan/2021 to 1/Apr/2021. **Ningxia** dataset features mountainous and hilly area with latitude range $34.5°\text{N}-42°\text{N}$ and longitude range $106°\text{E}-116°\text{E}$. There are 31×41 grids with grid size of 0.25 degree in latitude and longitude. Both NWP and observational weather data have hourly records including 8 weather factors from 1/Jan/2021 to 1/Jan/2022. In our experiments, we divide each dataset into training/validation/test subsets using a 7:1:2 ratio in chronological order. We define a temporal window of 7 time steps for NWP bias-correction, including the target time step, as well as the previous and next three steps.

4.2 Baselines and Experimental Settings

We compare WeatherGNN with three groups of methods. (1) Specific for bias-correction, including BiLSTM-T [Yang *et al.*, 2022]: using BiLSTM considering multiple covariates for bias-correction; HybridCBA [Han *et al.*, 2022]: combining CNN, BiLSTM, and attention mechanism to correct

[3] Since each row of $\prod_{i=0}^{l-1} \boldsymbol{S}^i$ only contains one 1 and others are 0s, the matrix multiplication of $\prod_{i=0}^{l-1} \boldsymbol{S}^i \tilde{\boldsymbol{H}}^{l}$ is implemented with indexing operation.

[4] The conclusion holds if we establish $\lfloor \log_k N \rfloor + 1$ levels, where k is the cluster size.

Dataset	Factor	Metric	NWP	BiLSTM-T	HybridCBA	ConvLSTM	AFNO	Swin	STGCN	HGCN	MegaCRN	WeatherGNN	ΔBaseline	ΔNWP
Ningbo	100ws	MAE	2.26	1.98	1.53	1.37	0.84	0.91	0.94	0.89	0.87	**0.80**	4.76%	64.60%
		RMSE	2.74	2.70	2.34	2.03	1.17	1.26	1.31	1.25	1.21	**1.14**	2.56%	58.39%
	10ws	MAE	1.38	1.37	1.18	0.91	0.58	0.67	0.65	0.63	0.61	**0.56**	3.45%	54.42%
		RMSE	1.71	1.69	1.65	1.28	0.81	0.87	0.91	0.88	0.85	**0.79**	2.47%	53.80%
	h	MAE	15.37	14.48	12.32	7.61	5.86	5.96	6.35	5.81	5.95	**5.63**	3.10%	63.37%
		RMSE	18.28	18.44	15.29	10.35	7.26	7.32	8.29	7.28	7.80	**7.12**	1.93%	61.05%
	2t	MAE	1.28	1.20	1.19	1.16	0.93	1.10	1.12	1.04	0.91	**0.90**	1.10%	29.69%
		RMSE	1.74	1.54	1.58	1.51	1.27	1.34	1.47	1.31	**1.23**	**1.23**	0.00%	29.31%
	tp	MAE	0.143	0.412	0.331	0.309	0.123	0.240	0.282	0.212	0.131	**0.115**	6.50%	19.58%
		RMSE	0.495	0.811	0.763	0.655	0.391	0.587	0.627	0.502	0.435	**0.339**	13.30%	31.52%
Ningxia	100ws(U)	MAE	2.93	2.69	2.33	2.22	2.00	2.12	2.33	2.08	1.95	**1.78**	8.72%	39.25%
		RMSE	3.83	3.41	2.99	2.87	2.58	2.81	3.01	2.69	2.54	**2.39**	5.91%	37.60%
	100ws(V)	MAE	3.39	2.79	2.67	2.51	2.25	2.30	2.51	2.29	2.27	**2.13**	5.33%	37.17%
		RMSE	4.52	3.59	3.43	3.24	2.93	3.11	3.22	3.03	2.94	**2.69**	8.19%	40.49%
	10ws(U)	MAE	1.79	1.65	1.44	1.40	1.22	1.22	1.52	1.33	1.23	**1.18**	3.28%	34.08%
		RMSE	2.38	2.16	1.87	1.84	1.60	1.64	1.97	1.73	1.60	**1.53**	4.38%	35.71%
	10ws(V)	MAE	2.03	1.73	1.55	1.50	1.32	1.41	1.56	1.38	1.35	**1.21**	8.33%	40.39%
		RMSE	2.74	2.24	2.02	1.96	1.75	2.08	2.01	1.85	1.80	**1.59**	9.14%	41.97%
	2t	MAE	2.81	2.71	2.72	2.44	2.27	2.27	2.54	2.20	2.28	**2.19**	0.45%	22.06%
		RMSE	3.74	3.49	3.54	3.15	2.91	2.89	3.27	**2.79**	2.92	2.80	-0.36%	25.13%

Table 1: Bias-correction Performance Comparison on Ningbo and Ningxia Datasets.

Factor		Ningbo 100ws		Ningxia 100ws(U)	
Metric		MAE	RMSE	MAE	RMSE
Variant	shared-F	0.83	1.18	1.87	2.47
	no-F	0.92	1.26	1.94	2.55
	geo-H	0.91	1.26	2.03	2.60
	met-H	0.89	1.25	1.92	2.57
	static-H	0.85	1.22	1.89	2.50
	no-H	0.96	1.31	2.33	2.82
WeatherGNN		**0.80**	**1.14**	**1.78**	**2.39**

Table 2: Ablation Study on 100m Wind Speed.

factors. (2) Vision methods used in many weather applications, including ConvLSTM [Shi *et al.*, 2015]: extending LSTM with convolutional gates; AFNO [Guibas *et al.*, 2021]: adopting Fourier neural operator to capture features adaptively; Swin [Liu *et al.*, 2021]: constructing a hierarchical transformer by a shifted windowing scheme. (3) GNN-based methods used in spatial-temporal applications, including STGCN [Yu *et al.*, 2018]: deploying graph convolution and temporal convolution to capture spatial and temporal dependencies; HGCN [Guo *et al.*, 2021]: operating hierarchical GNNs to utilize hierarchical spatial dependencies; MegaCRN [Jiang *et al.*, 2023]: introducing adaptive graphs to learn underlying spatial dependencies.

Baselines and WeatherGNN are implemented with Pytorch, executed on a server with one 32GB Tesla V100 GPU card, and well-tuned according to the performance on the validation set. The hyperparameter settings are summarized in Appendix. We optimize WeatherGNN using Adam optimizer with an initial learning rate of 0.003 and set the maximum number of epochs to 200. We halt training when the validation loss does not decrease for 15 consecutive epochs.

4.3 Performance Comparison

We use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to measure the performance of all methods. We present the results of common weather factors for main comparisons, e.g., wind speed (ws), pressure(h), temperature at 2m (2t), and total precipitation (tp). Table 1 shows the results, where **Bold** and Underline indicate the best and second best performance, respectively. Δ denotes the relative improvement between WeatherGNN and best baselines/NWP.

We can observe that: (1) NWP for specific areas can be significantly corrected, and WeatherGNN has an average improvement of over 41% compared with the original NWP. (2) Although baselines, especially AFNO and MegaCRN, have shown competitive performance, WeatherGNN achieves the **best performance** in 19 out of 20 cases across two datasets, outperforming the best baselines with average 4.50% on MAE and 4.75% on RMSE. Despite the remarkable improvement, WeatherGNN shows a slight inferiority in correcting 2t compared to the best baseline. The reason may be that the 2t factor varies slowly in time and space. The rapid changes in wind speeds may interfere with the correction of 2t due to the meteorological and spatial dependency modeling of WeatherGNN. (3) The bias-correction of WeatherGNN on wind speed is significant. Given the fact that the mechanism of wind formation is complex and wind speed changes over time and space rapidly, it is possible that the dynamic adjustment in the fast hierarchical GNN helps WeatherGNN adapt to weather dynamics.

4.4 Model Analysis

Ablation study. To evaluate the effectiveness of the factor GNN and fast hierarchical GNN of WeatherGNN, we conduct ablation studies on correcting the 100m wind speed of two datasets. For factor GNN, we design variants: (1) shared-F: only using the factor embedding to construct a shared factor graph for all grids, ignoring spatial heterogeneity; (2) no-F: replacing the factor graphs with a one-layer MLP to encode the weather information without considering the meteorological dependencies explicitly. For fast hierarchical GNN, we design variants: (3) geo-H: constructing the hierarchy only using geographical distances; (4) met-H: constructing the hierarchy only using DTW similarity of historical weather factor time series; (5) static-H: removing the dynamic adjustment and only adopting the static hierarchy when performing fast hierarchical message passing; (6) no-H: only using the $\boldsymbol{A}_{\text{spatial}}$ calculated by Eq. 6 to model spatial dependencies between grids without constructing the hierarchy.

All designs in WeatherGNN are proven to be effective based on the results in Table 2. We can observe that: (1) Terrain is crucial for local NWP bias-correction, even when

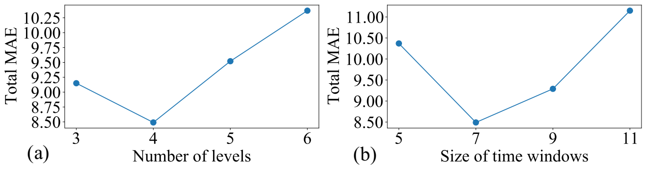

Figure 5: Results of Hyperparameter Study

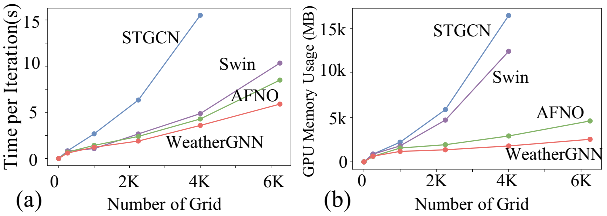

Figure 6: Model Efficiency Comparison.

it is solely used as input (as shown in ”no-F”). By comparing ”shared-F” and ”no-F”, we observe that modeling the dependencies between factors explicitly can improve the correction outcomes. Notably, making such dependencies adapt to the changing terrain can further enhance the correction effect, which validates the effectiveness of our factor GNN. (2) Constructing a hierarchical structure with geographical and meteorological distances significantly improves the bias-correction performance. The meteorological distance is found to be more effective than geographical distance, possibly because the influence of geography on weather factors is partly reflected in the weather factor sequences and DTW similarity can help to discover underlying spatial dependencies. Besides, the meteorological distance is helpful to unveil remote spatial dependencies. Furthermore, the hierarchical design integrating both factors achieves the best results.

Hyperparameter study. We adjust two hyperparameters, i.e., the number of levels (# levels) and the size of the time window, to investigate the effect of the static hierarchy and time length. We use total MAE of all factors in Ningbo dataset as the metric. We set # levels to 3, 4, 5, and 6. From Fig.5 (a), we find the correction performance is optimal when # level is 4. It is possible because smaller # levels fail to capture comprehensive multi-scale spatial dependencies, whereas larger # levels result in fewer grids in each supergrid, limiting the effectiveness of capturing local fine-grained dependencies. We set the size of the time window to 5, 7, 9, and 11. From Fig.5 (b), we find the time window is essential to the correction performance, with the optimal choice being 7. The possible reason is that smaller windows lack sufficient information to capture weather changes, while larger windows introduce excessive uncertainty about weather conditions.

Model efficiency. We evaluate the efficiency of WeatherGNN and baselines by increasing the number of grids and measuring running time and GPU memory usage. Fig. 6 illustrates that WeatherGNN (linear complexity) performs faster and requires less GPU memory than baselines. The efficiency advantages of WeatherGNN indicate good scalability, showing strong potential for larger regions, e.g., continental or global applications.

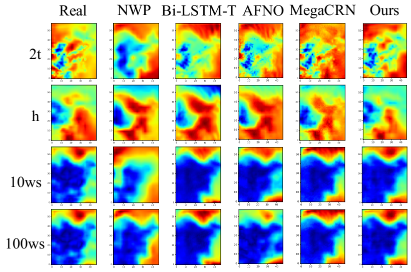

Figure 7: Visualization of Bias-Correction on Ningbo Dataset. Each row corresponds to a meteorological factor and each column corresponds to a method.

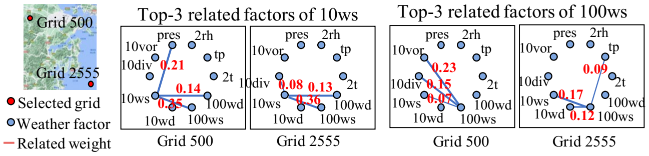

Figure 8: Illustration of Learned Factor Graphs of Two Grids on Two Different Types of Terrains.

Case study. We visualize an example at 22:00 6/Apr/2021 about origin NWP and corrected results of baselines and WeatherGNN. As shown in Fig. 7, NWP has a large deviation from the ground truth, while baselines and WeatherGNN can effectively reduce this discrepancy. In addition, Fig. 8 shows an example of the learned factor graphs selected randomly. Grid 500 locates in the mountains and Grid 2555 is in the ocean. 10ws and 100ws at both grids are closely related to each other. However, 10ws of Grid 500 is more relevant to pressure, as the mountainous regions have diverse terrain and are prone to rapid changes in air pressure, resulting in great effects on wind speeds. On the other hand, 10ws of Grid 2555 is affected by 10div (10m horizontal divergence, a measure of the local spreading or divergence of wind field in a horizontal plane at the height of 10 meters). One of the possible reasons is that on the sea horizon, the expansion of air in the horizontal direction has a significant impact on wind speed. More case studies about hierarchical structures, dynamic adjustment, and extreme weather are in Appendix.

5 Conclusion and Future Work

In this paper, we propose WeatherGNN, a GNN-based model that leverages meteorological and spatial dependencies under the guidance of domain knowledge for local NWP bias-correction. Specifically, we introduce a factor GNN and a fast hierarchical GNN to capture area-specific meteorological dependencies adaptively and dynamic spatial dependencies efficiently, respectively. Extensive experimental results demonstrate the superiority of WeatherGNN. In the future, we will investigate WeatherGNN for other applications, e.g., weather forecasting and downsampling. We also plan to apply WeatherGNN to larger real-world regions to investigate its scalability and robustness.

Contribution Statement

Binqing Wu and Weiqi Chen contributed equally to this work as co-first authors. Liang Sun and Ling Chen are the corresponding authors.

References

[Anselin, 2013] Luc Anselin. *Spatial econometrics: methods and models*, volume 4. Springer Science & Business Media, 2013.

[Bauer *et al.*, 2015] Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, September 2015.

[Bi *et al.*, 2023] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, pages 1–6, 2023.

[Chen *et al.*, 2023a] Kang Chen, Tao Han, Junchao Gong, Lei Bai, Fenghua Ling, Jing-Jia Luo, Xi Chen, Leiming Ma, Tianning Zhang, Rui Su, et al. Fengwu: Pushing the skillful global medium-range weather forecast beyond 10 days lead. *arXiv preprint arXiv:2304.02948*, 2023.

[Chen *et al.*, 2023b] Lei Chen, Xiaohui Zhong, Feng Zhang, Yuan Cheng, Yinghui Xu, Yuan Qi, and Hao Li. Fuxi: A cascade machine learning forecasting system for 15-day global weather forecast. *arXiv preprint arXiv:2306.12873*, 2023.

[Chen *et al.*, 2023c] Ling Chen, Jiahui Xu, Binqing Wu, and Jianlong Huang. Group-aware graph neural network for nationwide city air quality forecasting. *ACM Transactions on Knowledge Discovery from Data*, 18(3):1–20, 2023.

[Cui *et al.*, 2012] Bo Cui, Zoltan Toth, Yuejian Zhu, and Dingchen Hou. Bias correction for global ensemble forecast. *Weather and Forecasting*, 27(2):396–410, 2012.

[Delle Monache *et al.*, 2006] Luca Delle Monache, Thomas Nipen, Xingxiu Deng, Yongmei Zhou, and Roland Stull. Ozone ensemble forecasts: 2. a kalman filter predictor bias correction. *Journal of Geophysical Research: Atmospheres*, 111(D5), 2006.

[Durai and Bhradwaj, 2014] VR Durai and Rashmi Bhradwaj. Evaluation of statistical bias correction methods for numerical weather prediction model forecasts of maximum and minimum temperatures. *Natural Hazards*, 73:1229–1254, 2014.

[Esmaeilzadeh *et al.*, 2020] Soheil Esmaeilzadeh, Kamyar Azizzadenesheli, Karthik Kashinath, Mustafa Mustafa, Hamdi A Tchelepi, Philip Marcus, Mr Prabhat, Anima Anandkumar, et al. Meshfreeflownet: A physics-constrained deep continuous space-time super-resolution framework. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15. IEEE, 2020.

[Gao *et al.*, 2022] Zhihan Gao, Xingjian Shi, Hao Wang, Yi Zhu, Yuyang Bernie Wang, Mu Li, and Dit-Yan Yeung. Earthformer: Exploring space-time transformers for earth system forecasting. *Advances in Neural Information Processing Systems*, 35:25390–25403, 2022.

[Ge *et al.*, 2022] Tao Ge, Jaideep Pathak, Akshay Subramaniam, and Karthik Kashinath. Dl-corrector-remapper: A grid-free bias-correction deep learning methodology for data-driven high-resolution global weather forecasting. *arXiv preprint arXiv:2210.12293*, 2022.

[Gneiting and Raftery, 2005] Tilmann Gneiting and Adrian E Raftery. Weather forecasting with ensemble methods. *Science*, 310(5746):248–249, 2005.

[Guibas *et al.*, 2021] John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Efficient token mixing for transformers via adaptive fourier neural operators. In *International Conference on Learning Representations*, 2021.

[Guo *et al.*, 2021] Kan Guo, Yongli Hu, Yanfeng Sun, Sean Qian, Junbin Gao, and Baocai Yin. Hierarchical graph convolution network for traffic forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):151–159, 2021.

[Han *et al.*, 2021] Lei Han, Mingxuan Chen, Kangkai Chen, Haonan Chen, Yanbiao Zhang, Bing Lu, Linye Song, and Rui Qin. A deep learning method for bias correction of ecmwf 24–240 h forecasts. *Advances in Atmospheric Sciences*, 38(9):1444–1459, 2021.

[Han *et al.*, 2022] Yan Han, Lihua Mi, Lian Shen, CS Cai, Yuchen Liu, Kai Li, and Guoji Xu. A short-term wind speed prediction method utilizing novel hybrid deep learning algorithms to correct numerical weather forecasting. *Applied Energy*, 312:118777, 2022.

[Hess *et al.*, 2022] Philipp Hess, Markus Drüke, Stefan Petri, Felix M Strnad, and Niklas Boers. Physically constrained generative adversarial networks for improving precipitation fields from earth system models. *Nature Machine Intelligence*, 4(10):828–839, 2022.

[Hu *et al.*, 2019] Xiaodan Hu, Mohamed A Naiel, Alexander Wong, Mark Lamm, and Paul Fieguth. Runet: A robust unet architecture for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[Hu *et al.*, 2021] Shuai Hu, Yue Xiang, Da Huo, Shafqat Jawad, and Junyong Liu. An improved deep belief network based hybrid forecasting method for wind power. *Energy*, 224:120185, 2021.

[Jiang *et al.*, 2023] Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Toyotaro Suzumura, and Shintaro Fukushima. Megacrn: Meta-graph convolutional recurrent network for spatio-temporal modeling, 2023.

[Lam *et al.*, 2023] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, page eadi2336, 2023.

[Li *et al.*, 2022] Yu Li, Fei Tang, Xin Gao, Tongyan Zhang, Junfeng Qi, Jiarui Xie, Xinang Li, and Yuhan Guo. Numerical weather prediction correction strategy for short-term wind power forecasting based on bidirectional gated recurrent unit and xgboost. *Frontiers in Energy Research*, 9:1014, 2022.

[Liu *et al.*, 2021] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[Ma *et al.*, 2023] Guanghui Ma, Chunming Hu, Ling Ge, and Hong Zhang. Multi-view robust graph representation learning for graph classification. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 4037–4045, 2023.

[Miller, 2004] Harvey J Miller. Tobler's first law and spatial analysis. *Annals of the association of American geographers*, 94(2):284–289, 2004.

[Moran and Moran, 2009] Joseph M. Moran and Joseph M. Moran. *Weather Studies : Introduction to Atmospheric Science*. American Meteorological Association, Boston, MA, 4th edition, 2009.

[Müller, 2007] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[Nguyen *et al.*, 2023] Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. Climax: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343*, 2023.

[Pathak *et al.*, 2022] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[Rasp *et al.*, 2023] Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russel, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, et al. Weatherbench 2: A benchmark for the next generation of data-driven global weather models. *arXiv preprint arXiv:2308.15560*, 2023.

[Ravuri *et al.*, 2021] Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021.

[Shi *et al.*, 2015] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, page 802–810. MIT Press, 2015.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[Wu *et al.*, 2022] Lingfei Wu, Peng Cui, Jian Pei, Liang Zhao, and Xiaojie Guo. Graph neural networks: foundation, frontiers and applications. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4840–4841, 2022.

[Yang *et al.*, 2022] Xiaoying Yang, Shuai Yang, Mou Leong Tan, Hengyang Pan, Hongliang Zhang, Guoqing Wang, Ruimin He, and Zimeng Wang. Correcting the bias of daily satellite precipitation estimates in tropical regions using deep neural network. *Journal of Hydrology*, 608:127656, 2022.

[Yoshikane and Yoshimura, 2022] Takao Yoshikane and Kei Yoshimura. A bias correction method for precipitation through recognizing mesoscale precipitation systems corresponding to weather conditions. *PLoS Water*, 1(5):e0000016, 2022.

[Yu *et al.*, 2018] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 3634–3640, 2018.

[Zhang *et al.*, 2019] Tiejun Zhang, Pengcheng Yan, Zhaorong Li, Yousheng Wang, and Yaohui Li. Bias-correction method for wind-speed forecasting. *Meteorologische Zeitschrift*, 28(4):293–304, 11 2019.

[Zhang *et al.*, 2023a] Wei Zhang, Yueyue Jiang, Junyu Dong, Xiaojiang Song, Renbo Pang, Boyu Guoan, and Hui Yu. A deep learning method for real-time bias correction of wind field forecasts in the western north pacific. *Atmospheric Research*, 284:106586, 2023.

[Zhang *et al.*, 2023b] Yuchen Zhang, Mingsheng Long, Kaiyuan Chen, Lanxiang Xing, Ronghua Jin, Michael I Jordan, and Jianmin Wang. Skilful nowcasting of extreme precipitation with nowcastnet. *Nature*, 619(7970):526–532, 2023.

[Zhou *et al.*, 2020] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.