

Terrain-Guided Flatten Memory Network for Deep Spatial Wind Downscaling

Tingzhao Yu , Ruyi Yang, Yan Huang, Jinbing Gao, and Qiuming Kuang 

Abstract—High-resolution wind analysis plays an essential role in pollutant dispersion and renewable energy utilization. This article focuses on spatial wind downscaling. Specifically, a novel terrain-guided flatten memory network (abbreviated as TIGAM) with axial similarity constraint is proposed. TIGAM consists of three elaborately designed blocks, i.e., the similarity block, the reconstruction block, and the denoise block. To achieve long-spatial dependence, the similarity block interpolates low-resolution data to high resolution in an axial attention manner. Meanwhile, the reconstruction block aims to obtain a clearer high-resolution representation in closed form. Taking both of the meteorological prior and network design principle into consideration, this article also proposes a flatten memory module with learnable input for high-resolution denoising. Furthermore, for accurate detail reconstruction, a terrain-guided enhanced loss is presented benefitting from the high-resolution remote sensing data. This loss function integrates wind spatial distribution and terrain elegantly. Extensive quantitative and qualitative experiments demonstrate the superiority of the proposed TIGAM.

Index Terms—Deep learning, flatten memory network, image super-resolution, meteorological methods, spatial wind downscaling.

I. INTRODUCTION

ACCURATE and fine meteorological forecasts have always been urgent needs for both scientific research and intelligent service. Thanks to the development of remote sensing techniques, researchers are capable of obtaining high-resolution observations. The high computation costs of numerical weather simulations, however, limit the availability of fine-scale meteorological predictions. As a consequence, spatial downscaling, which is a hopeful technique of generating high-resolution meteorological data from the low-resolution one, grows into a hot research topic among atmospheric research. As an important component of meteorological analysis [1], [2], [3], [4], [5], spatial wind downscaling [6], [7] plays a substantial role among areas including oceanographic research [8], [9], climate analysis [10], renewable energy generation [11], and accurate meteorological forecasting [12].

Manuscript received 13 July 2022; revised 5 September 2022 and 11 October 2022; accepted 21 October 2022. Date of publication 31 October 2022; date of current version 9 November 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62106270, and in part by the Innovation Program of Public Meteorological Service Center under Grant M2021002. (Corresponding author: Tingzhao Yu.)

The authors are with the Public Meteorological Service Center, China Meteorological Administration, Beijing 100081, China (e-mail: tsingzao@hotmail.com; ryyangPMSC@hotmail.com; yanhuangpmc@hotmail.com; jingbinggaoPMSC@hotmail.com; qmkuang@hotmail.com).

Digital Object Identifier 10.1109/JSTARS.2022.3218016

Spatial downscaling aims to provide meteorological reanalysis/forecast as precise as possible for given areas, which has been studied for decades. Various downscaling methods have been proposed to improve spatial resolutions in meteorological research, which can be categorized into dynamical downscaling and statistical downscaling [13]. For dynamical downscaling, high-resolution numerical models are employed to simulate subgrid-scale physical processes based on the large-scale circulation predicted from coarser-scale models [14], [15]. However, dynamical downscaling is well known for its high requirements on computing resources. In contrast, statistical methods can produce competitive results with a low-cost computational resource compared to dynamical downscaling. There emerge a large amount of statistical methods to enhance the spatial resolutions of meteorological variables, such as bilinear interpolation, nearest neighbor [16], support vector machines [17], and weather generators [18]. For example, Jia et al. [19] develop a multiple linear regression model for downscaling the spatial precipitation fields by applying the relationships between precipitation and other environmental factors such as topography and vegetation. Lima et al. [20] propose a Bayesian Kriging model to downscale daily Global Climate Model rainfall into a fine-resolution grid, which can successfully reproduce the spatial variability in the observed rainfall. Moreover, based on a nonlinear relationship between precipitation at high-resolution and covariates at coarse/fine resolution, adaptable random forests have been utilized for spatial precipitation downscaling [21]. Empirical analysis indicates that it can outperform bilinear interpolation and replicate the spatial and temporal distribution of observed precipitation fields. Recently, Jing et al. [22] propose an end-to-end network, which consists of a global cross-attention module, a multifactor cross-attention module, and a residual convolutional module, for satellite precipitation downscaling. Wang et al. [23] propose a new algorithm based on the Taylor expansion for land surface temperature downscaling. Results show that this method got the best downscaled results when the land surface temperature acquired time is consistent with the time of empirical concavity factor.

As for spatial wind downscaling, the diagnostic models, such as California Meteorology (CALMET) [24], is commonly used to generate high-resolution wind fields with horizontal resolution of several hundred meters [25], [26], [27]. Höhle et al. [6] use a convolutional neural network (CNN)-based model for downscaling low-resolution wind forecast simulations to a higher spatial resolution. Kirchmeier et al. [28] propose a probabilistic-based statistically downscaling approach

to predict a daily-varying probability distribution of local-scale wind speed, conditioned on the large-scale wind speed. Moreover, wind downscaling is extremely complicated on complex topography. Helbig et al. [29] apply local, fine-scale topographic parameters to surface wind speed downscaling method for numerical weather prediction (NWP) simulation. Based on coarse-scale NWP wind speeds, the method can efficiently reproduce distribution statistics in the near-surface wind speed measurements. Winstral et al. [30] introduce a novel optimization scheme that can capture the local terrain structure for gridded wind speed downscaling. Given the wind-terrain interaction, the downscaling method effectively improves the error metrics and spatial distributions correlation between the downscaled wind field and observation. Based on the CNN, Dujardin and Lehning [31] propose a near-surface wind fields downscaling approach that considers the state of the atmosphere on various scales and its interaction with high-resolution topography. The downscaling method perform well in generating 50-m resolution wind fields, especially under the effects of complex topography like ridge acceleration, sheltering, and deflection.

Recent advances in machine learning, especially deep learning, have witnessed great success for meteorological applications. Specifically, the widely investigated image super-resolution in the field of computer vision shares a similar objective with meteorological spatial downscaling, i.e., minimizing the reconstruction loss as far as possible. Recently, many super-resolution methods based on deep learning have achieved state-of-the-art performance. Superresolution convolutional neural network (SRCNN) [32] is the first deep learning method for single image super-resolution. It is a three-layer CNN for patch extraction, nonlinear mapping, and high-resolution reconstruction. Basically, the SRCNN preprocesses the low-resolution image with bicubic interpolation, while the fast super-resolution CNN (FSRCNN) [33] applies deconvolution to reconstruct the corresponding high-resolution image from the low-resolution one directly. Recently, the efficient subpixel CNN (ESPCN) [34] has become the common upsampling strategy and it has been widely used in state-of-the-art methods, e.g., enhanced deep residual super-resolution (EDSR) [35], residual channel attention network (RCAN) [36], hybrid residual attention network (HRAN) [37], and super-resolution recursive fractal network (SRRFN) [38]. There are also methods trying to dig out the deeper nonlinear information with more convolutional layers, e.g., very deep super-resolution network (VDSR) [39]. Specifically, different convolutional layers extract distinct features, in order to fully exploit these information, Liu et al. [40] raise a residual feature aggregation framework for more efficient feature extraction. This framework groups several residual modules together and directly forwards the features on each local residual branch by adding skip connections. Therefore, it is capable of aggregating these informative residual features to produce more representative features. However, different from the model-based methods that can handle image super-resolution with different scale factors under a unified framework, the learning-based methods generally lack such flexibility. To address this issue, Zhang et al. [41] propose an end-to-end trainable unfolding

network that leverages both learning-based methods and model-based methods. By unifying diverse scale factors into a joint end-to-end framework, image super-resolution can be divided into several subproblems, that can be solved iteratively. Recent contributions [42], [43], [44], [45] also start to seeking a better performance under the widely used transformer [46] mechanism. Apart from convolutional methods, there are brilliant pioneer contributions focusing on recurrent super-resolution. Most of them devote to video super-resolution [47], [48], [49], because the convolutional recurrent networks are inherently suitable for spatial-temporal data. There are also researches concentrating on recurrent single-image super-resolution. For example, Yang et al. [50] introduce a deep-edge-guided recurrent residual network to progressively recover the high-frequency details. Yang et al. [51] propose a deep recurrent fusion network. Through which, multilevel features with large receptive-field can be obtained. Han et al. [52] propose a dual-state recurrent network for single-image super-resolution, and proved many state-of-the-art super-resolution techniques can be reformulated as a single-state recurrent network.

Regardless of the resemblance between meteorological downscaling and image super-resolution, we claim that utilizing the method designed for super-resolution directly for downscaling confronts with the following four *ASCL* challenges that must be thoroughly considered.

- 1) *Apriority*: Atmospheric science has been investigated for hundreds of years, and it has now established a perfect theory with multiple physical-constrained differential equations, including the motion equation, the continuity equation, the energy equation, the state equation, and the potential temperature equation. Through solving these equations, the current and future meteorological state can be obtained. In other words, the causality of meteorological downscaling is embedded into prior expert knowledge. However, it is not the case for image super-resolution. From the perspective of optimization, image super-resolution is an ill-posed problem with multiple feasible solutions. It is also a challenging direction of imposing prior knowledge into the data-driven model, i.e., the image super-resolution methods. Consequently, in order to take the benefits of the existing image super-resolution techniques, the apriority must be taken into consideration.
- 2) *Similarity*: Spatially, the meteorological state at a particular location is highly correlated with its corresponding neighbors. Temporally, driven by the earth rotation and revolution, the same meteorological state can occur periodically. More importantly, the meteorological state at the same latitude or longitude often tends to be consistent. For a natural image, taking a cat as example, it can appear at anywhere and anytime of the image, without much apriority. Nevertheless, it is not the case for meteorological data. For example, the typhoon can only occur at the ocean or coastal areas during regular time. This means, for accurate spatial downscaling, the high similarity correlation must be handled elaborately. From the perspective of computer vision, the axial attention mechanism [53], which explores

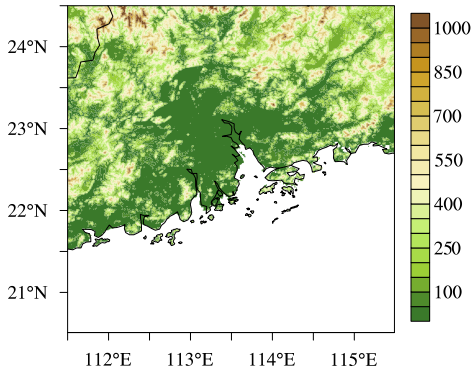


Fig. 1. Study area (20.51°N–24.50°N, 111.50°E–115.49°E). The corresponding high-resolution topography is also embedded to TIGAM for high-resolution spatial wind reconstruction.

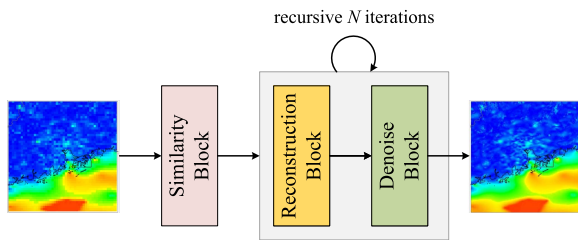


Fig. 2. Illustration of the proposed TIGAM. Intuitively, TIGAM is comprised of three basic blocks, i.e., the similarity block for coarse reconstruction, the reconstruction block for clear reconstruction, and the denoise block for clean reconstruction.

long-range spatial dependencies can provide an insightful solution to this problem.

- 3) *Coupling*: Even though the relationship among different meteorological elements is extremely complex, they have great impact to each other. Taking the wind as a simple example, the pressure difference p between two locations caused wind w . Both of the temperature T and the air mass are key impact factors of p . In addition, the humidity h influences air mass a lot. The wind w in turn, can also change h and T . In order to tackle this challenge, a common practice is to impose additional weather variables as input. However, treating weather factors as input data directly has several imperfections. On one hand, due to the fact that weather factors are typically obtained via professional meteorological sensors or observation stations, it is impractical and costly to record extra weather data. On the other hand, it is somewhat unreasonable to integrate weather factors together directly. In other words, different weather factors vary in data distribution, and how to integrate these factors together remains a worthwhile research. For generality, a substitutable latent learnable input with spatial memory should make sense.
- 4) *Locality*: Distinct from most of the computer vision related tasks, the meteorological problem is typically scale susceptible. Typically, a *reasonable* RGB image might be colorful and uniform with feasible saturation and contrast on a large scale for human perception. However, being affected by the terrain, the extreme weather, e.g., tornado,

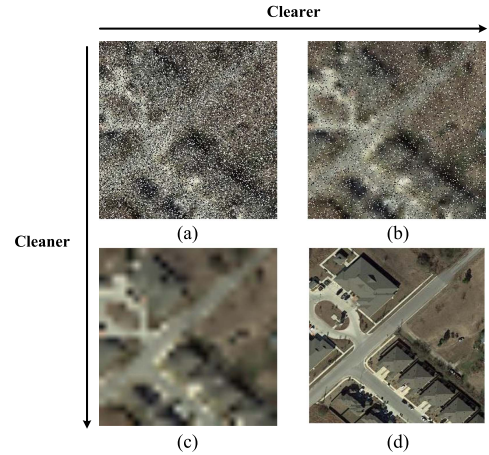


Fig. 3. Illustration about the concepts “cleaner” and “clearer.”

thunder, hail, etc., often occurs locally on a relatively small scale. For better performance, the high-resolution terrain should be a key indicator of local meteorological variation.

To address the former illustrated issues, this article proposes terrain-guided flatten memory network (TIGAM), an elaborately designed deep convolutional network with meteorological prior guidance for spatial wind downscaling. Specifically, taking the atmospheric apriority into consideration, this article proposes a novel flatten memory module with learnable position-sensitive input. Benefitting from this mechanism, TIGAM is capable of simulating the prior meteorological motion equations and depicting the multielement coupling implicitly. Spatially, the axial attention policy is embedded into TIGAM for better describing the long-range spatial similarity. Moreover, through a comprehensive statistical analysis, we find that the winds are highly correlated to terrain. Consequently, this article further presents a new enhanced loss. This auxiliary guidance, which calculates the standard deviation among local areas and treats the standard deviation of terrain as targets, restricts its spatial distribution analogous to the corresponding terrain. Experiments compared with three basic interpolation techniques and 20 state-of-the-art deep learning methods demonstrate the effectiveness of the proposed TIGAM.

The remainder of this article is structured as follows. Section II presents the study area and the used data. The proposed method, i.e., TIGAM, is illustrated in detail in Section III. Section IV formulates the experimental configuration and the results. Details are discussed in Section V, while conclusions are finally drawn in Section VI.

II. MATERIALS

A. Study Areas

The study region is constituted of the central and southern Guangdong Province of China and part of South China Sea, with a good coverage of 20.51°N–24.50°N, 111.50°E–115.49°E (see Fig. 1). The topography of this area exhibits spatial variability, with an arc-shaped mountain ranges (e.g., Julian Mountain) in

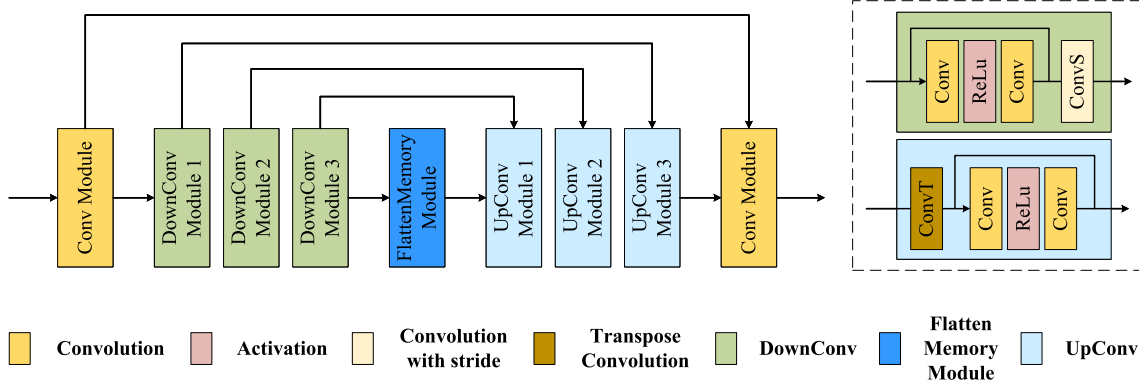


Fig. 4. Employed meteorological prior guided dense block. This block consists of four types of modules, i.e., the basic conv module, the downconv module, the flatten memory module, and the upconv module. Given a reconstructed wind image X^R , the dense block works in a cascaded ResUNet manner. Specifically, a flatten memory module is designed to fit the atmospheric differential equation, i.e., (8), in an implicit way.

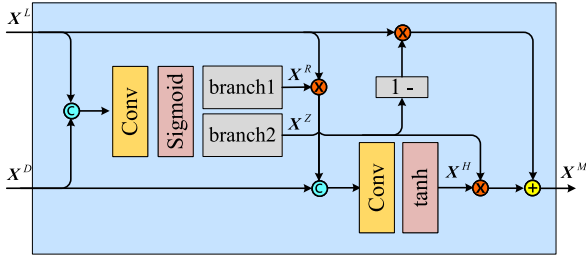


Fig. 5. Flatten memory module. To imitate the multifactor atmospheric theory, an extra learnable input X^L is adopted. Intuitively, the flatten memory module can be regarded as a spatial GRU network, and that is the reason why we called this module a flatten memory module.

the north, a plain area in the middle, and the ocean area in the south. The wind field in this area is extremely affected by the interaction of complex topography, land–sea distribution, and weather systems (such as typhoons, low-level jets, cold fronts, etc.), making spatial wind downscaling in this region a worth studying problem.

B. Data Description

This article evaluates the proposed method on hourly China Atmospheric Real-time Analysis System—surface analysis at 1-km spatial resolution (CARAS-SUR1km) reanalysis data. CARAS-SUR1km is a refined real-time professional meteorological product generation system developed by the Public Meteorological Service Center, China Meteorological Administration. It provides surface real-time professional service products (temperature, relative humidity, u-wind, v-wind, etc.) per hour with $1 \text{ km} \times 1 \text{ km}$ spatial resolution. This system integrates more than 60 000 ground stations observation data, CMA-GRAPES regional typhoon prediction model (GRAPES-TYM) data, high-precision terrain and underlying surface data, and other multisources observation data, by using the upgraded multigrid variational method. Therefore, the CARAS-SUR1km products have good temporal and spatial continuity, and can reflect the local refined topographic features well. Following the typical strategy for single image super-resolution in computer vision, the high-resolution CARAS-SUR1km data are treated

as ground truth and it is sampled to 2 km as low-resolution model input. The high-resolution (1 km) orography data are also employed as auxiliary guidance information. Specifically, the CARAS-SUR1km data from 2016 to 2018 are used for training, while the year 2019 is employed for validation and the year 2020 for testing.

C. Data Preprocessing

For better convergence, the wind components are preprocessed utilizing standard normalization as

$$X = \frac{X_{\text{org}} - \text{mean}(X_{\text{org}})}{\text{std}(X_{\text{org}})} \quad (1)$$

while for terrain, it is first divided by 1500, and then, normalized as (1). Following the typical normalization trick in deep convolutional networks, the *global* mean and standard deviation are computed.

III. METHODS

A. Problem Formulation

This article focuses on spatial wind downscaling. Different from precipitation or temperature, winds are vectors with both magnitude v and direction ω . For better understanding, winds are typically decomposed into two orthogonal components, i.e., the U-wind $v \cos \omega$ and the V-wind $v \sin \omega$. Analogous to color images, this article regards these two elements as two relevant channels. Without specific illustration, this two-channel data are referred to *wind image*.

Accordingly, given a low-resolution wind image $X_t \in \mathbb{R}^{w \times h \times c}$ of a specific area at time t , where w and h represent the spatial width and height, $c = 2$ denotes the two orthogonal components, spatial wind downscaling aims at recovering a relatively high-resolution and accurate wind image $f(\theta; X_t, s) = \hat{Y}_t \in \mathbb{R}^{sw \times sh \times c}$, which is as similar as possible to the ground-truth wind image $Y_t \in \mathbb{R}^{sw \times sh \times c}$. Here, s represents the downscaling ratio and f denotes the downscaling function defined by parameter θ , which can be mathematically obtained

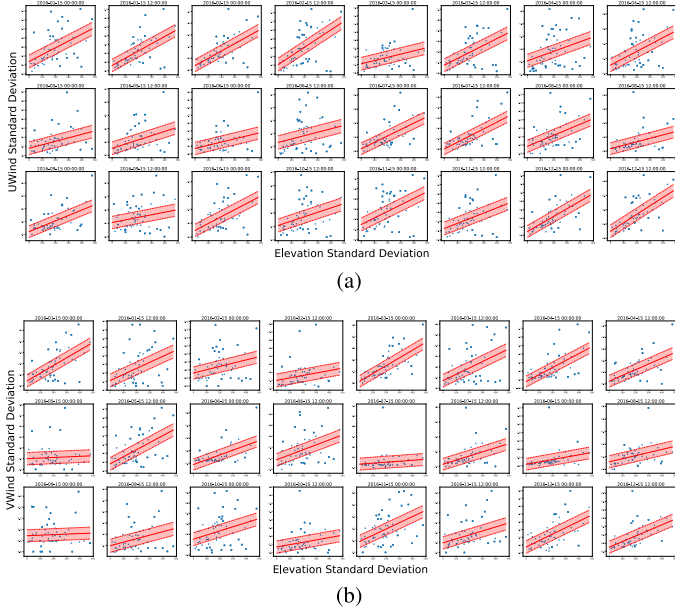


Fig. 6. Wind standard deviation versus terrain standard deviation. To be specific, the high-resolution terrain and the corresponding wind image are first split into small nonoverlap patches. Then, the standard deviations are calculated within each patch. Furthermore, a linear regression model is raised to fit the two standard deviations. Even though this relationship cannot be described in a fixed-parameter model, these two figures indicate that the wind standard deviation is highly correlated to terrain standard deviation.

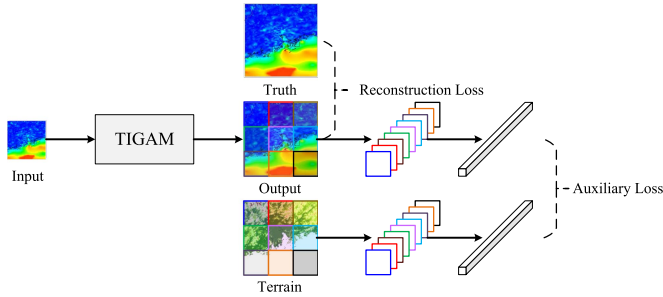


Fig. 7. Mechanism of the terrain-guided enhanced loss. Except the reconstruction loss, an auxiliary terrain-guided loss, which restricts the relationship between terrain and wind spatial distribution, is employed.

via

$$\theta = \arg \min_{\theta} \sum_t \mathcal{L}(f(\theta; X_t, s), Y_t) \quad (2)$$

where \mathcal{L} is a specified loss function, such as L1 loss, L2 loss, or others.

B. TIGAM Architecture

To achieve spatial wind downscaling, this article proposes TIGAM network. An intuitive illustration can be found in Fig. 2. Basically, TIGAM consists of three types of blocks, i.e., the similarity block, the reconstruction block, and the denoise block.

Given a low-resolution wind image, the similarity block first interpolates it to the desired size, and then, employs axial attention to discover the long-range spatial dependence.

After that, the reconstruction block aims to find a clearer high-resolution wind image and the denoise block devotes to obtain a cleaner high-resolution one. We should also note that, inspired by deep unfold super-resolution network (USRNET) [41], the reconstruction block, and the denoise block are executed iteratively in an RNN manner for better performance. We then present details of each block.

Analysis: Taking an RGB image as example, a “clearer” image emphasizes more on clarity with details, while a “cleaner” image highlights on purity without noise. Fig. 3 presents an intuitive expression. In Fig. 3, the subfigure (a) is noisy and blurred, the subfigure (b) is clear but noisy, the subfigure (c) is clean but blurred, and the subfigure (d) is clear and clean. These two processes are typically necessary. To be specific, given a low-resolution input, the reconstruction block aims to reconstruct a relatively high-resolution one. Nevertheless, this process introduces noise inevitably due to the inner property of solving an ill-posed super-resolution (downscaling) problem. These two types of block are then required.

C. Similarity Block—Exploring Long-Range Spatial Similarity

As illustrated in Section III-B, the similarity block interpolates a low-resolution wind image X_t to high resolution roughly by taking long-range spatial dependence into consideration, through the axial attention mechanism as [53]. Without loss of generality, we omit the subscript t for simplicity (i.e., X is equivalent to X_t without specific illustration). Consequently, this block can be formulated mathematically as

$$\begin{aligned} X' &= \text{super sample}_s(X) \\ &\Downarrow \\ \begin{cases} q, k, v, r = W_h X' \\ z_{i,j}^h = \text{softmax}_{a \in \mathcal{N}(i)} (q_{i,j}^T k_{a,j} + q_{i,j}^T r_{a-i}^q + k_{a,j}^T r_{a-i}^k) \\ X'' = \sum_{a \in \mathcal{N}(i)} z_{i,j}^h (v_{a,j} + r_{a-i}^v) \end{cases} \\ &\Downarrow \\ \begin{cases} q, k, v, r = W_w X'' \\ z_{i,j}^w = \text{softmax}_{b \in \mathcal{N}(j)} (q_{i,j}^T k_{i,b} + q_{i,j}^T r_{b-j}^q + k_{i,b}^T r_{b-j}^k) \\ X^S = \sum_{b \in \mathcal{N}(j)} z_{i,j}^w (v_{i,b} + r_{b-j}^v) \end{cases} \quad (3) \end{aligned}$$

where q , k , v , and r are the corresponding query, key, value, and relative position encoding, respectively. $\text{supersample}_s(\dots)$ represents interpolating with scale factor s , and \mathcal{N} denotes the corresponding neighbors. $X^S \in \mathbb{R}^{sw \times sh \times c}$ is the output of the similarity block.

Analysis: Let us denote $w' = sw$, $h' = sh$, and $C = c$ for simplicity, i.e., $X' \in \mathbb{R}^{w' \times h' \times C}$. When conducting width-axis attention, i.e., calculating q , k , and v using X' , the size of learnable weight matrixes are typically set as $W_{hq} \in \mathbb{R}^{C \times d}$, $W_{hk} \in \mathbb{R}^{C \times d}$, and $W_{hv} \in \mathbb{R}^{C \times 2d}$, where d is the hidden dimension. Taking q as example, the width-axis multiplication operation is then executed on two matrixes $X'_i \in \mathbb{R}^{h' \times C}$ and $W_{hq} \in \mathbb{R}^{C \times d}$ for each $i = 1, \dots, w'$. And the resulting matrix is $q_i = X'_i W_{hq} \in \mathbb{R}^{h' \times d}$. The same operation is implemented

TABLE I
OVERALL PERFORMANCE OF THE PROPOSED TIGAM AND OTHER STATE-OF-THE-ART METHODS

Method	MAE			SSIM			PSNR		
	U	V	Mean	U	V	Mean	U	V	Mean
Bilinear	0.154	0.175	0.165	0.831	0.832	0.832	47.627	46.370	46.999
Bicubic	0.159	0.180	0.170	0.833	0.834	0.834	47.376	46.083	46.730
Nearest	0.162	0.184	0.173	0.791	0.792	0.792	45.682	44.470	45.076
SRCNN (2014)	0.127	0.141	0.134	0.881	0.886	0.884	49.189	47.917	48.553
SubPixel (2016)	0.152	0.198	0.175	0.820	0.775	0.798	48.188	46.372	47.280
VDSR (2016)	0.159	0.180	0.170	0.833	0.834	0.834	47.376	46.083	46.730
FSRCNN (2016)	0.181	0.246	0.214	0.796	0.785	0.791	47.118	45.747	46.433
EDSR (2017)	0.130	0.169	0.150	0.870	0.854	0.862	49.641	47.489	48.565
SRDenseNet (2017)	0.056	0.060	0.058	0.956	0.958	0.957	55.694	54.957	55.326
LAPSRN (2017)	0.058	0.061	0.060	0.950	0.955	0.953	54.255	53.893	54.074
RDN (2018)	0.072	0.053	0.063	0.954	0.967	0.961	55.150	56.374	55.762
RCAN (2018)	0.059	0.061	0.060	0.952	0.957	0.955	55.018	54.754	54.886
DBPN (2018)	0.080	0.099	0.090	0.924	0.917	0.921	51.922	50.464	51.193
RNAN (2019)	0.083	0.097	0.090	0.926	0.923	0.925	51.518	50.049	50.784
SRFBN (2019)	0.050	0.051	0.051	0.959	0.966	0.963	55.587	55.475	55.531
SAN (2019)	0.038	0.041	0.040	0.975	0.976	0.976	59.157	58.588	58.873
USRNET (2020)	0.036	0.038	0.037	0.977	0.978	0.978	59.514	58.991	59.253
RFANet (2020)	0.075	0.077	0.076	0.926	0.936	0.931	52.011	51.873	51.942
CSNLA (2020)	0.077	0.086	0.082	0.925	0.926	0.926	51.781	50.665	51.223
NLSN (2021)	0.074	0.072	0.073	0.931	0.944	0.938	52.739	52.620	52.680
SwinIR (2021)	0.057	0.059	0.058	0.953	0.958	0.956	54.680	54.275	54.478
WindTopo* (2022)	0.034	0.039	0.037	0.957	0.958	0.957	58.215	57.017	57.571
Restormer (2022)	0.032	0.033	0.033	0.981	0.983	0.982	60.234	59.777	60.006
TIGAM (ours)	0.031	0.033	0.032	0.982	0.983	0.983	60.521	59.994	60.258

on k and v . In practice, we often employ an integrated learnable weight matrix

$$W_h = \text{cat}([W_{hq}, W_{hk}, W_{hv}]) \in \mathbb{R}^{C \times 4} \quad (4)$$

for efficiency. In other words

$$[q_i, k_i, v_i] = X'_i[W_{hq}, W_{hk}, W_{hv}] = X'_i W_h. \quad (5)$$

At last, for all $i = 1, \dots, w'$, we denote

$$q, k, v = W_h X'. \quad (6)$$

As for the relative position encoding r , it is a learnable input vector predefined in the axial attention layer that can be treated as relevant to the input data, i.e., X' . Through former illustration, the proposed similarity block can exploit long-range spatial dependence, i.e., the *similarity* challenge, via the axial attention mechanism.

D. Reconstruction Block—Recovering a Clearer Wind Image

Intuitively, even though the similarity block takes both of latitude and longitude relevance into consideration through axial attention, the reconstructed wind image X^S is relatively coarse. Following the strategy implemented in USRNet [41], this article also decouples the reconstruction process [i.e., (2)] into two cascaded half-quadratic subproblems. The first subproblem, which attempts to recover a clearer wind image X^R in a closed-form solution, can be solved utilizing fast Fourier transform without optimizing any trainable parameters. The second subproblem, i.e., the denoising problem, which focuses on achieving a cleaner wind image \hat{Y} , can be settled via the following proposed denoise block with a flatten memory strategy.

E. Denoise Block—Reconstructing a Cleaner Wind Image

The denoise block is designed in a ResUNet framework. A detailed description can be found in Fig. 4. Specifically, the denoise block contains a Basic Conv Module for feature transformation, three DownConv modules for high-level feature extraction, a flatten memory module for fitting atmospheric prior, three UpConv modules for feature reconstruction, and a basic Conv module for wind image reconstruction.

For efficiency, the basic Conv module contains only a single convolutional layer without any normalization or activation, i.e., $X^B = \text{Conv}(X^R)$. The DownConv module consists of three convolutional layers with a ReLU activation and a residual connection. Note that the last convolutional layer is set to be with stride for down sampling. The DownConv module can be formulated as

$$\begin{cases} X^0 = \text{ReLU}(W_0 * X^B) \\ X^1 = W_1 * X^0 + X^S \\ X^2 = W_2 *_S X^1 \end{cases} \quad (7)$$

where $*$ represents convolution with parameter W and $*_S$ denotes convolution with stride. We omit the bias term for simplicity.

Atmospherically, the wind is highly correlated with many essential factors as

$$\underbrace{\frac{d\vec{V}}{dt} = -\frac{1}{\rho} \nabla p + \vec{g} + \vec{N} - 2\vec{\Omega} \wedge \vec{V}}_{\text{learnable term}} \quad (8)$$

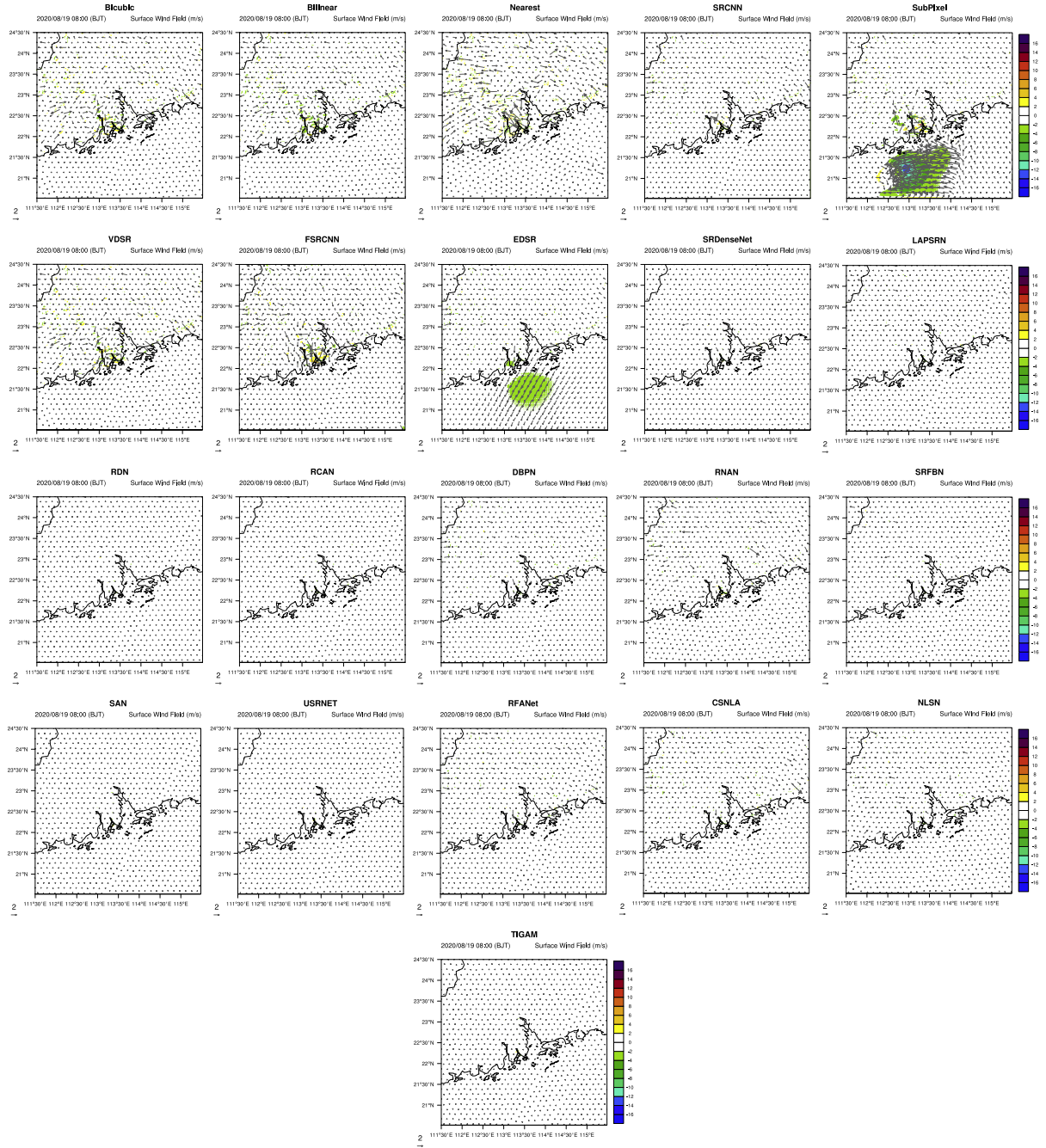


Fig. 8. Visualization of the differences between the downscaling results and the ground truth. Except wind speed differences (color areas), the wind direction discrepancies are also shown in the figure using directed arrow. Better zoomed in and viewed in color.

where \vec{V} , p , ρ , $\vec{\Omega}$, \vec{g} , and \vec{N} indicate wind, air pressure, air density, Coriolis force, gravitational acceleration, and air friction, respectively. Therefore, according to (8), the interactive influence among different meteorological elements must be thoroughly considered for accurate spatial wind downscaling.

Specifically, a novel flatten memory module is proposed to deal with this issue. Suppose the output of last DownConv module is X^D , the flatten memory module fit (8) and takes an extra learnable input X^L to imitate other meteorological factors.

Generally, X^D and X^L are first concatenated and processed via a single convolutional layer with sigmoid activation, i.e.,

$$\begin{pmatrix} X^Z \\ X^R \end{pmatrix} = \text{sigmoid} (W_{ZR} * [X^L; X^D]). \quad (9)$$

After that, we obtain a hidden memory representation X^H via

$$X^H = \tanh (W_H * [X^R X^L; X^D]). \quad (10)$$

TABLE II
ABLATION STUDIES OF DIFFERENT BLOCKS INCLUDING THE AXIAL SIMILARITY BLOCK (AXIAL), THE FLATTEN MEMORY BLOCK (FM), AND THE TERRAIN-GUIDED ENHANCED LOSS (TL)

Method	MAE			SSIM			PSNR		
	U	V	Mean	U	V	Mean	U	V	Mean
SRCNN (Baseline)	0.127	0.141	0.134	0.880	0.881	0.881	49.189	47.917	48.553
SRCNN + Axial	0.122 ↓	0.133 ↓	0.128 ↓	0.881 ↑	0.886 ↑	0.884 ↑	49.850 ↑	48.671 ↑	49.261 ↑
SRCNN + TL	0.115 ↓	0.128 ↓	0.122 ↓	0.888 ↑	0.885 ↑	0.887 ↑	49.566 ↑	48.282 ↑	48.924 ↑
SRCNN + FM	0.097 ↓	0.109 ↓	0.103 ↓	0.901 ↑	0.911 ↑	0.906 ↑	50.982 ↑	49.667 ↑	50.325 ↑
SwinIR (Baseline)	0.057	0.059	0.058	0.953	0.958	0.956	54.680	54.275	54.478
SwinIR + Axial	0.057	0.059	0.058	0.954 ↑	0.958	0.956	54.806 ↑	54.500 ↑	54.653 ↑
SwinIR + TL	0.041 ↓	0.044 ↓	0.043 ↓	0.972 ↑	0.975 ↑	0.974 ↑	58.245 ↑	57.727 ↑	57.986 ↑
SwinIR + FM	0.056 ↓	0.056 ↓	0.056 ↓	0.957 ↑	0.964 ↑	0.961 ↑	55.519 ↑	55.289 ↑	55.404 ↑

TABLE III
SENSITIVE ANALYSIS CONSIDERING THE TRADEOFF PARAMETER USING SWINIR AS BASELINE

δ	MAE			SSIM			PSNR		
	U	V	Mean	U	V	Mean	U	V	Mean
0	0.057	0.059	0.058	0.953	0.958	0.956	54.680	54.275	54.478
0.01	0.041	0.042	0.042	0.971	0.975	0.973	58.006	57.770	57.888
0.05	0.041	0.044	0.043	0.972	0.975	0.974	58.245	57.727	57.986
0.1	0.059	0.062	0.061	0.949	0.954	0.952	54.229	53.795	54.012
0.5	0.052	0.055	0.054	0.959	0.960	0.960	55.533	54.744	55.139

TABLE IV
NUMBER OF PARAMETERS AND RUNNING TIME FOR ALL COMPARED METHODS (THE BLUE VALUES DENOTE TOP-3 PARAMETER / TIME CONSUMPTION)

Method	CSNLA	DBPN	EDSR	FSRCNN	LAPSRN	NLSN	SwinIR
#Param	0.77M	23.31M	1.91M	0.09M	3.26M	6.71M	54.43M
#Time	40ms	30ms	5ms	5ms	22ms	24ms	30ms
Method	RCAN	RDN	RFANET	RNAN	SAN	SRCNN	Restormer
#Param	49.43M	86.60M	0.78M	1.86M	61.88M	0.08M	60.01M
#Time	43ms	40ms	6ms	34ms	208ms	3ms	102.31ms
Method	SRDENSENET	SRFBN	SUBPIXEL	USRNET	VDSR	WindTopo	TIGAM
#Param	8.18M	2.37M	0.24M	66.50M	0.59M	43.05M	64.30M
#Time	11ms	7ms	3ms	70ms	4ms	26ms	36ms

The final output of the flatten memory module is defined as

$$X^M = (1 - X^Z)X^L + X^Z X^H. \quad (11)$$

More details can also be found in Fig. 5.

The UpConv module recovers high-level feature representation in a cascaded mechanism. Mathematically, it can be written as

$$\begin{cases} X^0 = W_0 *_{\mathcal{T}} (X^M + X^D) \\ X^1 = \text{ReLU}(W_1 * X^0) \\ X^2 = W_2 * X^1 + X^0 \end{cases} \quad (12)$$

where $*_{\mathcal{T}}$ denotes transpose convolution.

Denote the output of last UpConv Block as X^U , another basic convolutional layer is implemented to reconstruct the high-resolution wind image as $\hat{Y} = \text{Conv}(X^U)$.

Here, X^L represents the learnable input, X^D denotes the output of last DownConv module, and X^M stands for the output of flatten memory module. As for X^Z , X^R , and X^H , we follow their initial declarations in GRU [54] or ConvGRU [55] as update gate, reset gate, and hidden state.

Analysis: On one hand, atmospheric science has been investigated for hundreds of years, and it has now established a perfect theory. On the other hand, the emergency of deep learning has facilitated many related areas, such as meteorology. We believe that the brilliant meteorological theory is a constructive guidance for deep-learning-based meteorological methods. Then, we attempt to explore an elaborate network combing with the established meteorological theory. Consequently, the flatten memory module is proposed.

Specifically, the flatten memory module is designed to deal with the *apriority* and *coupling* challenges. In the case of apriority, it is hard to design a deep network under the physical constraint [see (8)] directly. However, (8) can be roughly divided into two parts, i.e., wind and other variables. To imitate (8), two input branches corresponding to wind and other variables are essential. For simplicity and generality, we defined other variables a learnable term. Putting this term aside, (8) contains only basic operations such as addition and multiplication. A single convolutional layer is then capable of imitating this relationship. Nevertheless, instead of using a single layer network, we employ a memory unit to fit this function. Meanwhile, the



Fig. 9. Sensitivity analysis about the number of denoise blocks.

coupling challenge is considered through the former illustrated learnable input.

The reason why we define it a flatten memory module is twofold. *First*, this module is mainly inspired by the gated recurrent unit (GRU), which consists of multiple temporal memory units for sequential learning. The basic structure illustrated in Fig. 5 is similar to this memory unit. That is the reason why we still say it a *memory* module. *Second*, instead of exploiting temporal dependence, the newly proposed module trying to explore spatial relevance. This equals to flatten a recurrent module spatially while still preserves the property of memory, even though this memory stores spatial relevance rather than temporal dependence. That is the reason why we say it a *flatten* module. Consequently, we say this a flatten memory module.

In spite of the fact that ConvGRU and the proposed flatten memory module share same basic architecture, these two components are distinct at cell level. ConvGRU cell takes the original stable sequential snapshots as input while the flatten memory module cell operates on learnable spatial dimension. More specifically, the only input for ConvGRU is indeed the given sequence X . While for the flatten memory module, apart from the given input X^D , there is an additional learnable position-sensitive part X^L . This learnable part is independent from X^D . We say it learnable owing to the fact that X^L is obtained via backward optimization instead of forward computation. We say it position sensitive because the size of X^L is similar as X^D . From this point of view, the ConvGRU cell is a special case of the flatten memory module cell. In other words, when X^L is set to be frozen during backward optimization and be relative to X^D in the forward process, the flatten memory module degrades to ConvGRU.

F. Terrain-Guided Enhanced Loss

Through the former illustrated strategy, it is capable of dealing with first three ASCL challenges (Apriority, Similarity, and Coupling) and obtaining a relatively high-resolution wind image. As for the Locality, we observe that the standard deviation of wind within local areas is highly relevant to local terrain. Fig. 6 reveals this discovery.

Specifically, after an investigation of early contributions and a statistical analysis, we found that the standard deviation of local terrain is an important indicator for the clear wind image reconstruction. Therefore, a terrain-guided enhanced loss is elaborately designed (see Fig. 7).

To be specific, the reconstructed wind image \hat{Y} is first split into nonoverlap windows of size $k \times k$, the standard deviation is then calculated within each window. The same operations are also conducted on terrain T . The enhanced loss is then define as

$$\mathcal{L}_{en} = \left\| \sum \text{std}(\hat{Y}_{k \times k}) - \xi \sum \text{std}(T)_{k \times k} \right\|^p \quad (13)$$

ξ is a hyperparameter and the final loss \mathcal{L} is a combination of both the reconstruction loss $\mathcal{L}_{re} = \|\hat{Y} - Y\|^q$ and the enhanced loss \mathcal{L}_{en} , i.e.,

$$\mathcal{L} = \mathcal{L}_{re} + \delta \mathcal{L}_{en} \quad (14)$$

where δ is a hyperparameter for tradeoff. The involved parameters can be optimized via back-propagation.

Analysis: According to early works [29], [30], [31] and after a statistical analysis, we say there is high correlation between terrain and wind, especially between terrain std and wind std. Unfortunately, how this relevance is mathematically formulated is unknown and has not been revealed. To find out this correlation, we denote $x = \text{std}(\hat{Y}_{k \times k})$ and $y = \text{std}(T_{k \times k})$, where \hat{Y} , T , and k represent the reconstructed high-resolution wind image, the terrain and the nonoverlap window size, respectively. We can define this correlation abstractly via a nonlinear map as $y = f(x)$. According to the theory of Taylor expansion, we can reformulate this as

$$\begin{aligned} y &= f(x) \\ &= f(x_0) + f'(x_0)(x - x_0) + R_n \\ &= f'(x_0)x + f(x_0) - f'(x_0)x_0 + R_n \\ &= f'(x_0)x + C. \end{aligned} \quad (15)$$

Alternatively, we rewrite (15) as $x - \frac{1}{f'(x_0)}y = \frac{C}{f'(x_0)}$. This meets with the proposed terrain-guided enhanced loss (13). We must note that this enhanced loss is quite coarse and limited. However, this is supposed to be a reasonable approximation of $f(x)$ and the experimental results also demonstrate its validity.

IV. RESULTS

A. Experiment Setting

TIGAM is implemented on Pytorch. Codes and sample data are available at <https://github.com/Tsingzao/TIGAM>. Generally, the downscale factor is set to be 2. The numbers of reconstruction and denoise block are all set to be 8. Without specific illustration, the hidden channel is set to be 64 and the convolutional kernel size is 3×3 . There are three DownConv and three UpConv modules in the denoise block, the nonoverlap window size $k = 8$ and the hyperparameter $\delta = 0.05$.

For model training, Adam with learning rate $1e^{-4}$ is employed and training stops after 20 epochs or the validation loss does not decrease after five epochs. The batchsize is set to be 8.

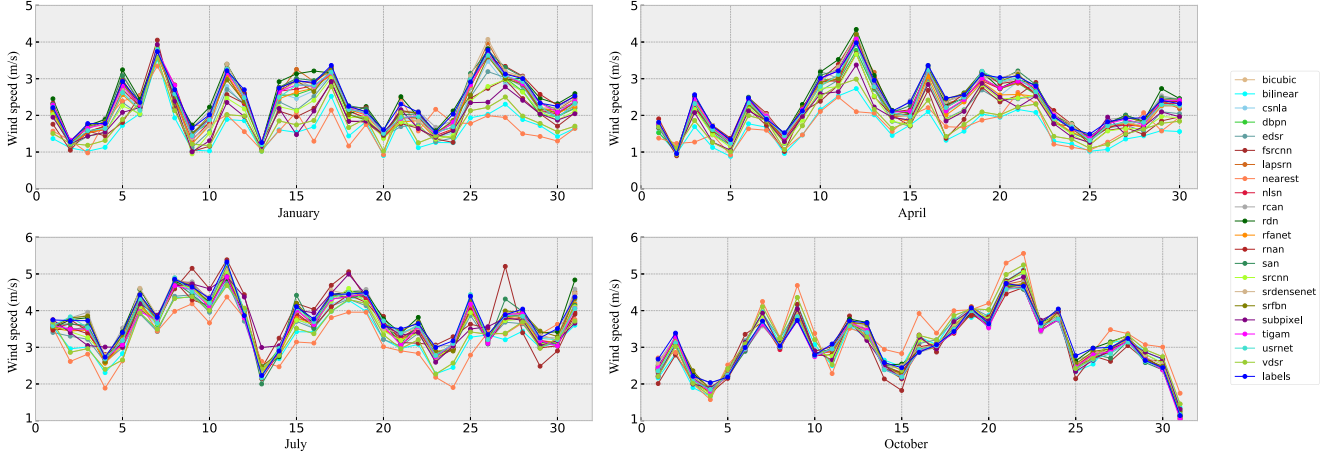


Fig. 10. Daily maximum wind at the central study area during January, April, July, and October. Large discrepancies can be found during January, while the difference during October is the smallest.

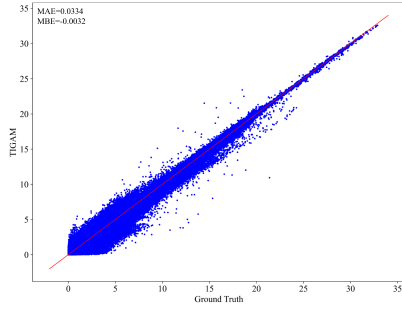


Fig. 11. Illustration of the bias between TIGAM and ground truth.

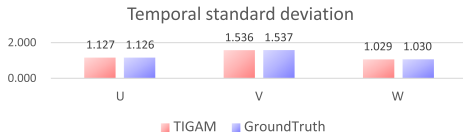


Fig. 12. Illustration about the temporal deviation.

To verify the superiority of the proposed TIGAM, 20 kinds of methods including both of traditional interpolation methods such as bilinear, bicubic, and nearest interpolation, and advanced deep learning methods such as SRCNN [32], Sub-Pixel [34], VDSR [39], fast super-resolution CNN (FSRCNN) [33], EDSR [35], SRDenseNet [56], LAPSARN [57], residual dense network (RDN) [58], RCAN [36], DBPN (Deep Back-Projection Network) [59], RNAN (Residual Non-local Attention Network) [60], SRFBN (Super Resolution FeedBack Network) [61], SAN (Second-order Attention Network) [62], USRNET (Unfold Super Resolution Network) [41], RFANet (Residual Feature Aggregation Network) [40], CSNLA (Cross-Scale Non-Local Attention network) [63], NLSN (Non-Local Sparse Attention network) [64] SwinIR [44], Restormer [45], WindTopo [31], etc., are discussed in detail. Specifically, the following two categories of experiments are conducted.

- 1) *Quantitative comparison*: To demonstrate the superiority of TIGAM, quantitative comparisons are first provided in Section IV-C.
- 2) *Qualitative comparison*: Intuitive comparisons among the proposed TIGAM and other state-of-the-art super-resolution algorithms are then presented in Section IV-D.

B. Evaluation Metrics

For fair comparison, this article employs standard metrics mean absolute error, for quantitative evaluation. Suppose Y_t and \hat{Y}_t denote the t th ground-truth high-resolution wind image and the down-scaled one, then the corresponding mathematical formulation are defined as follows:

$$\text{MAE} = \frac{1}{T} \sum_{t=1}^T \|Y_t - \hat{Y}_t\|. \quad (16)$$

Apart from that, the widely used structure similarity (SSIM) and peak signal noise ratio (PSNR) are also evaluated. Specifically, SSIM and PSNR are formulated as

$$\text{SSIM}_t = \frac{(2\mu_{Y_t}\mu_{\hat{Y}_t} + c_1)(2\sigma_{Y_t\hat{Y}_t} + c_2)}{(\mu_{Y_t}^2 + \mu_{\hat{Y}_t}^2 + c_1)(\sigma_{Y_t}^2 + \sigma_{\hat{Y}_t}^2 + c_2)}$$

$$\text{PSNR}_t = 10 \times \log_{10} \frac{I_{\max}^2}{\text{RMSE}_t^2} \quad (17)$$

where I_{\max} denotes the maximum value of wind, c_1 and c_2 are constants, and μ and σ represents the corresponding mean and variance, respectively. For PSNR, the related RMSE is defined as $\text{RMSE} = \sqrt{\frac{1}{T} \sum_{t=1}^T \|Y_t - \hat{Y}_t\|^2}$.

For MAE, the smaller the better, while for SSIM and PSNR, the larger the better.

C. Quantitative Comparison

This subsection shows quantitative comparisons considering MAE, SSIM, and PSNR. Results can be found in Table I.

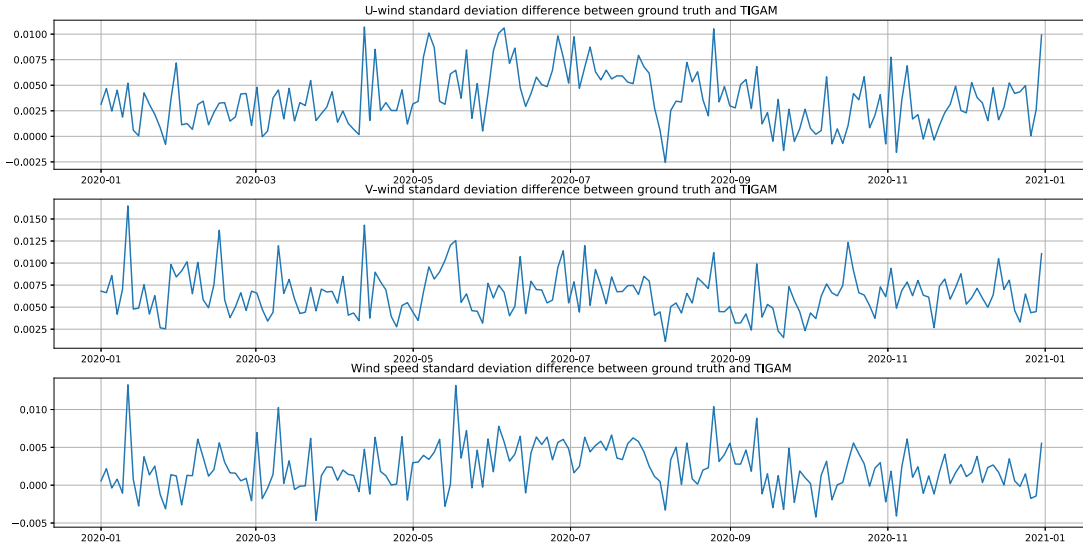


Fig. 13. Illustration about the spatial deviation.

Specifically, SRCNN [32] is the first proposed deep super-resolution method with only three convolutional layers. Then, VDSR [39] and FSRCNN [33] improve the SRCNN with more convolutional layers. SubPixel [34] reconstructs high-resolution image without any interpolation or transpose convolution. To get more specific feature representations, SRDenseNet [56], RDN [58], and RFANet [40] employ residual connection as basic block. The widely used attention mechanism is also exploited in EDSR [35], RCAN [36], RNAN [60], SAN [62], CSNLA [63], NLSN [64] etc. Other representational methods include LAPS RN [57], DBPN [59], SRFBN [61], and USRNET [41]. SwinIR [44] is a recently proposed dedicated image super-resolution structure based upon Swin Transformer [65]. WindTopo [31] is a recently proposed deep-learning-based wind downscaling method using high-resolution topograph. Apart from that, Restormer [45] is a recently proposed transformer architecture especially designed for high-resolution image restoration, such as denoising, deblurring, and deraining. Image super-resolution can be also treated as a specific task of restoration. Consequently, implementing restormer for wind downscaling is reasonable.

More concretely, early contributions such as SRCNN [32], SubPixel [34], VDSR [39], FSRCNN [33], and EDSR [35] perform slightly worse than the other methods, and this is mainly due to the fact that these models are comprised of simple stacked convolutional layers without much specific design for downscaling. Obviously, the recent methods SAN [62], USRNET [41], and the newly proposed TIGAM are superior to the remaining methods. WindTopo [31] outperforms most of the single image super-resolution methods and the performance of Restormer [45] is quite competitive to TIGAM. From this table, it should be noted that the proposed TIGAM outperforms other methods consistently on all of the evaluated metrics (i.e., MAE, SSIM, and PSNR) in terms of both U and V components, demonstrating the effectiveness of the proposed method. We should note that WindTopo is an elaborately designed downscaling network for particular data format. We cannot employ this for our data directly. Therefore, we modify the original released

codes on five aspects. First, we change the station-based data format to grid based. Second, the reimplemented WindTopo employs nonlocal convolution instead of multiple zoom patch-based convolution for efficiency. Third, the network is altered to totally pixel-in pixel-out. Fourth, for grid-based downscaling, we conduct the most widely used pixelshuffle operation. Fifth, limited by the available data, we only take the u-wind, v-wind, DEM, slope, and aspect as model input.

D. Qualitative Results

In order to give a more intuitive comparison, this subsection presents qualitative comparisons among TIGAM and other state-of-the-art methods. Specifically, the results at time stamp 2020-8-19 08:00 (BJT) are showed, during which, the seventh typhoon Higos in 2020 attacked the study area. The maximum wind speed reaches up to 126 km/h, and caused huge damages to the passed areas.

Fig. 8 exhibits the differences between downscaling results and the ground truth. Since the proposed terrain-guided enhanced loss has taken the terrain effect into consideration, it enables our model to achieve a better spatial wind downscaling performance in the terrain area. It is worth noting that other meteorological variables beyond wind are also highly correlated with the terrain, e.g., temperature and precipitation. Therefore, our proposed model can be extended to these meteorological variables. This is a challenging topic and would be explored in our future research.

V. DISCUSSION

A. Ablation Study

Ablation studies are first conducted to systemically and comprehensively analyze TIGAM. Table II demonstrates the performance of the newly proposed blocks, including the axial similarity block (Axial), the flatten memory strategy (FM), and the terrain-guided enhanced loss (TL). For efficiency, both of SRCNN and SwinIR are employed as the baselines. SRCNN/SwinIR + * denotes adding the corresponding * block to SRCNN.

According to Table II, the proposed three blocks have positive effects, illustrating that the newly proposed strategies reasonable for spatial wind downscaling. On one hand, the axial similarity block promotes the downscaling results due to the fact that meteorological elements are highly correlated spatially. On the other hand, the local spatial distribution constraint, which is embedded into the terrain-guided enhanced loss, also facilitates spatial downscaling. Finally, benefitting from the flatten memory strategy, a position-sensitive learnable input is taken into account and it can fit the motion (8) indirectly.

In addition, implementing SwinIR as the baseline outperforms that of SRCNN. The reasons are straightforward for that SwinIR consists of more elaborately designed modules for feature extraction, feature transformation, and details reconstruction. The proposed three new blocks promote model performance when they are embedded to SwinIR, illustrating these blocks feasible for wind downscaling. However, we should also note that the performance improvement of employing SwinIR as baseline is smaller than using the SRCNN.

B. Sensitive Analysis

As illustrated in Section III-B, the denoise block is executed iteratively in an RNN manner. Therefore, there is indeed only one denoise block. However, the memory units such as LSTM often runs in a recursive manner and the flatten memory module can be regarded as a special case of memory units. Consequently, the denoise block working in an iterative manner is reasonable. We then do sensitive analysis about the number of recurrent blocks. The results are presented in Fig. 9. From this figure, with the growth of iterative denoise block, the model performance first increases, and then, drops. Obviously, the model efficiency also decreases with the growth of iterative denoise block. Taking both of the model performance and efficiency into consideration, we select eight blocks eventually.

We also do sensitive analysis considering the tradeoff parameter δ using SwinIR as the baseline model. The results are presented in Table III. Specifically, the model achieves relatively best performance when the tradeoff parameter is set to be 0.05.

C. Maximum Wind

Furthermore, Fig. 10 presents daily maximum wind during January, April, July, and October, which can represent the winter, spring, summer, and autumn, respectively. The results indicates that the TIGAM can successfully reproduce the temporal variability in the observed daily maximum field. The daily maximum wind speed obtained by TIGAM is basically consistent with the observation. Compared to other methods, TIGAM has a better or competitive performance in terms of the bias of daily maximum wind in spatial wind downscaling.

D. Bias Analysis

We present an intuitive expression about the wind speed bias in Fig. 11. From this figure, the results of TIGAM exhibit high correlation with the ground truth, especially in extreme winds. The mean bias error (MBE) between TIGAM and observed

winds is -0.0032 , which indicates that TIGAM slightly underestimated the wind speed compared to the ground truth.

E. Deviation Analysis

Regression-based downscaling approaches have a tendency to underestimate extremes limited by the frequently used L1 or L2 loss. To avoid these underestimations and for better describing extreme details, we proposed the terrain-guided enhanced loss. Figs. 12 and 13 present the local deviations. Specifically, we display these deviations in two manners, i.e., temporally and spatially.

For temporal deviation, we select the central point of the study area as example and calculate its standard deviation temporally (from 2020-01-01 00:00 clock to 2020-12-31 23:00 clock). Similarly, there is no significant difference between the ground-truth observation and the TIGAM prediction, indicating that TIGAM performs consistently with time.

For spatial deviation, we calculate the standard deviation among the study area hourly. In order to illustrate TIGAM's performance clearly, Fig. 13 shows the deviation difference between ground-truth observation and TIGAM prediction. From Fig. 13, there is no remarkable difference between them, illustrating TIGAM is a feasible method for extreme spatial downscaling.

F. Complexity Analysis

Table IV shows the reimplemented model complexity and running time of all compared methods. Specifically, the running time is calculated on a single 24 G TITAN Xp GPU. We should note that the running time and model complexity are not linearly correlated. Compared with the simple architectures such as SRCNN, FSRCNN, SUBPIXEL, and VDSR, both of the model complexity and running time of TIGAM is large. Compared with the relatively complex methods such as RCAN, RDN, SAN, and USRNET, both of the model complexity and running time of TIGAM is comparable. Taking the model complexity, the running time and the model performance all into consideration, TIGAM is superior to other methods.

VI. CONCLUSION

This article proposed a novel TIGAM network for spatial wind downscaling. Specifically, a similarity block with an axial attention mechanism is proposed for exploiting long-range spatial dependence. To investigate the advanced meteorological prior, this article also presents a denoise block with a flatten memory module. Benefitting from this module, a position-sensitive input can be learned to fit the complex influence among different meteorological elements. Furthermore, a terrain-guided enhanced loss is proposed to depict the details of downscaling results. Qualitative and quantitative results demonstrate the superiority of the proposed TIGAM.

Generally, spatial downscaling equals to generate unknown values conditioned on limited information, and the widely explored generative adversarial networks provides a possible solution. Consequently, the future work will focus on learning a deep generative model for meteorological downscaling.

Besides, in view of the strong temporal dependence among different meteorological elements, such as temperature, relative humidity, air pressure, and winds, a multimodal spatial-temporal downscaling algorithm with strict atmospheric restriction is also under construction. Apart from that, this article tries to solve the deep-learning-based atmospheric problem indirectly. We leave the direct method our future research topic.

REFERENCES

- [1] H. Gao et al., "A spatio-temporal neural network for fine-scale wind field nowcasting based on Lidar observation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote. Sens.*, vol. 15, pp. 5596–5606, Jul. 2022.
- [2] W. Wang et al., "All-weather near-surface air temperature estimation based on satellite data over the Tibetan plateau," *IEEE J. Sel. Topics Appl. Earth Observ. Remote. Sens.*, vol. 15, pp. 3340–3350, Mar. 2022.
- [3] C. Radhakrishnan, V. Chandrasekar, S. C. Reising, and W. K. Berg, "Rainfall estimation from TEMPEST-D cubesat observations: A machine-learning approach," *IEEE J. Sel. Topics Appl. Earth Observ. Remote. Sens.*, vol. 15, pp. 3626–3636, Apr. 2022.
- [4] C. Guo, W. Ai, S. Hu, X. Du, and N. Chen, "Sea surface wind direction retrieval based on convolution neural network and wavelet analysis," *IEEE J. Sel. Topics Appl. Earth Observ. Remote. Sens.*, vol. 15, pp. 3868–3876, May 2022.
- [5] T. Yu, Q. Kuang, J. Zheng, and J. Hu, "Deep precipitation downscaling," *IEEE Geosci. Remote. Sens. Lett.*, vol. 19, Jan. 2022, Art. no. 1001405.
- [6] K. Höhle, M. Kern, T. Hewson, and R. Westermann, "A comparative study of convolutional neural network models for wind field downscaling," *Meteorological Appl.*, vol. 27, no. 6, 2020, Art. no. e1961.
- [7] J. Reynolds, M. P. Clarizia, and E. Santi, "Wind speed estimation from CYGNSS using artificial neural networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote. Sens.*, vol. 13, pp. 708–716, Feb. 2020, doi: [10.1109/JSTARS.2020.2968156](https://doi.org/10.1109/JSTARS.2020.2968156).
- [8] A. Hossain, M. M. Jacob, and W. L. Jones, "Ocean vector wind retrievals from TRMM using a novel combined active/passive algorithm," *IEEE J. Sel. Topics Appl. Earth Observ. Remote. Sens.*, vol. 13, pp. 5569–5579, Sep. 2020, doi: [10.1109/JSTARS.2020.3023720](https://doi.org/10.1109/JSTARS.2020.3023720).
- [9] X. Ye, M. Lin, Q. Song, and Q. Wang, "The optimized small incidence angle setting of a composite Bragg scattering model and its application to sea surface wind speed retrieval," *IEEE J. Sel. Topics Appl. Earth Observ. Remote. Sens.*, vol. 13, pp. 1248–1257, Mar. 2020, doi: [10.1109/JSTARS.2020.2974660](https://doi.org/10.1109/JSTARS.2020.2974660).
- [10] Y. Xu, "Estimates of changes in surface wind and temperature extremes in southwestern Norway using dynamical downscaling method under future climate," *Weather Climate Extremes*, vol. 26, 2019, Art. no. 100234.
- [11] M. Oh, J. Lee, J.-Y. Kim, and H.-G. Kim, "Machine learning-based statistical downscaling of wind resource maps using multi-resolution topographical data," *Wind Energy*, vol. 25, no. 6, pp. 1121–1141, 2022.
- [12] J. Barcons, M. Avila, and A. Folch, "A wind field downscaling strategy based on domain segmentation and transfer functions," *Wind Energy*, vol. 21, no. 6, pp. 409–425, 2018.
- [13] R. Wilby and T. Wigley, "Downscaling general circulation model output: A review of methods and limitations," *Prog. Phys. Geogr., Earth Environ.*, vol. 21, no. 4, pp. 530–548, 1997.
- [14] C. L. Castro, R. A. Pielke Sr., and G. Leoncini, "Dynamical downscaling: Assessment of value retained and added using the regional atmospheric modeling system (RAMS)," *J. Geophys. Res., Atmospheres*, vol. 110, no. D5, 2005, Art. no. D05108.
- [15] W. Zhang, J. Harff, R. Schneider, M. Meyer, and C. Wu, "A multiscale centennial morphodynamic model for the southern Baltic coast," *J. Coastal Res.*, vol. 27, no. 5, pp. 890–917, 2011.
- [16] S. Gangopadhyay, M. Clark, and B. Rajagopalan, "Statistical downscaling using k-nearest neighbors," *Water Resour. Res.*, vol. 41, no. 2, 2005, Art. no. W02024.
- [17] S. Tripathi, V. Srinivas, and R. S. Nanjundiah, "Downscaling of precipitation for climate change scenarios: A support vector machine approach," *J. Hydrol.*, vol. 330, no. 3, pp. 621–640, 2006.
- [18] D. S. Wilks, "Stochastic weather generators for climate-change downscaling, Part II: Multivariable and spatially coherent multisite downscaling," *WIREs Climate Change*, vol. 3, no. 3, pp. 267–278, 2012.
- [19] S. Jia, W. Zhu, A. Lü, and T. Yan, "A statistical spatial downscaling algorithm of TRMM precipitation based on NDVI and DEM in the Qaidam basin of China," *Remote Sens. Environ.*, vol. 115, no. 12, pp. 3069–3079, 2011.
- [20] C. H. Lima, H.-H. Kwon, and Y.-T. Kim, "A Bayesian Kriging model applied for spatial downscaling of daily rainfall from GCMs," *J. Hydrol.*, vol. 597, 2021, Art. no. 126095.
- [21] X. He, N. W. Chaney, M. Schleiss, and J. Sheffield, "Spatial downscaling of precipitation using adaptable random forests," *Water Resour. Res.*, vol. 52, no. 10, pp. 8217–8237, 2016.
- [22] Y. Jing, L. Lin, X. Li, T. Li, and H. Shen, "An attention mechanism based convolutional network for satellite precipitation downscaling over China," *J. Hydrol.*, vol. 613, 2022, Art. no. 128388.
- [23] S. Wang, Y. Luo, M. Li, K. Yang, Q. Liu, and X. Li, "A Taylor expansion algorithm for spatial downscaling of MODIS land surface temperature," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 5002717-1–5002717-17, Jun. 2022, Art. no. 5002717.
- [24] J. S. Scire, F. R. Robe, M. E. Fernau, and R. J. Yamartino, "A User Guide for the CALMET Meteorological Model," vol. 37, Earth Tech, Concord, MA, USA, 2000.
- [25] S. Tang, S. Huang, H. Yu, M. Gu, and J. Tang, "Impact of horizontal resolution in CALMET on simulated near-surface wind fields over complex terrain during super typhoon Meranti (2016)," *Atmospheric Res.*, vol. 247, 2021, Art. no. 105223.
- [26] R. S. Jose, J. L. Perez, J. Pecci, A. Garzon, and M. Palacios, "Climate downscaling processes using WRF/chem and CALMET modelling systems: Madrid case study," in *Proc. Int. Conf. Utility Cloud Comput.*, 2014, pp. 830–835.
- [27] D. O. Oshurok, "Evaluation of the CALMET mesoscale meteorological model against wind measurements in four landscape zones of Ukraine," in *Proc. Int. Conf. Young Scientists Energy Issues*, 2018, pp. 1-96-1-104.
- [28] M. C. Kirchmeier, D. J. Lorenz, and D. J. Vimont, "Statistical downscaling of daily wind speed variations," *J. Appl. Meteorol. Climatol.*, vol. 53, no. 3, pp. 660–675, 2014.
- [29] N. Helbig, R. Mott, A. Van Herwijnen, A. Winstral, and T. Jonas, "Parameterizing surface wind speed over complex topography," *J. Geophys. Res., Atmospheres*, vol. 122, no. 2, pp. 651–667, 2017.
- [30] A. Winstral, T. Jonas, and N. Helbig, "Statistical downscaling of gridded wind speed data using local topography," *J. Hydrometeorol.*, vol. 18, no. 2, pp. 335–348, 2017.
- [31] J. Dujardin and M. Lehning, "Wind-topo: Downscaling near-surface wind fields to high-resolution topography in highly complex terrain with deep learning," *Quart. J. Roy. Meteorological Soc.*, vol. 148, no. 744, pp. 1368–1388, 2022.
- [32] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2014, vol. 8692, pp. 184–199.
- [33] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comput. Vis.*, 2016, vol. 9906, pp. 391–407.
- [34] W. Shi et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1874–1883.
- [35] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1132–1140.
- [36] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, vol. 11211, pp. 294–310.
- [37] A. Muqet, M. T. B. Iqbal, and S. Bae, "HRAN: Hybrid residual attention network for single image super-resolution," *IEEE Access*, vol. 7, pp. 137020–137029, 2019.
- [38] J. Li, Y. Yuan, K. Mei, and F. Fang, "Lightweight and accurate recursive fractal network for image super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2019, pp. 3814–3823.
- [39] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1646–1654.
- [40] J. Liu, W. Zhang, Y. Tang, J. Tang, and G. Wu, "Residual feature aggregation network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2356–2365.
- [41] K. Zhang, L. V. Gool, and R. Timofte, "Deep unfolding network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3214–3223.
- [42] H. Chen et al., "Pre-trained image processing transformer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12299–12310.
- [43] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, "Learning texture transformer network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5790–5799.

[44] J. Liang, J. Cao, G. Sun, K. Zhang, L. V. Gool, and R. Timofte, "SWINIR: Image restoration using swin transformer," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2021, pp. 1833–1844.

[45] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5728–5739.

[46] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[47] J. Chen, Y. Yuan, J. Zhang, and J. Luo, "Self-guided video super-resolution based on a fast deformable convGRU model," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2022, pp. 1–6.

[48] S. López-Tapia, A. Lucas, R. Molina, and A. K. Katsaggelos, "Gated recurrent networks for video super resolution," in *Proc. Eur. Signal Process. Conf.*, 2021, pp. 700–704.

[49] Y. Huang, W. Wang, and L. Wang, "Video super-resolution via bidirectional recurrent convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 1015–1028, Apr. 2018.

[50] W. Yang et al., "Deep edge guided recurrent residual learning for image super-resolution," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5895–5907, Dec. 2017.

[51] X. Yang et al., "DRFN: Deep recurrent fusion network for single-image super-resolution with large factors," *IEEE Trans. Multimedia*, vol. 21, no. 2, pp. 328–337, Feb. 2019.

[52] W. Han, S. Chang, D. Liu, M. Yu, M. Witbrock, and T. S. Huang, "Image super-resolution via dual-state recurrent networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1654–1663.

[53] T. Yu, Q. Kuang, and R. Yang, "ATMconvGRU for weather forecasting," *IEEE Geosci. Remote. Sens. Lett.*, vol. 19, pp. 1003805-1–1003805-5, Sep. 2021.

[54] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*.

[55] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving deeper into convolutional networks for learning video representations," 2015, *arXiv:1511.06432*.

[56] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 4809–4817.

[57] W. Lai, J. Huang, N. Ahuja, and M. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5835–5843.

[58] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2472–2481.

[59] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1664–1673.

[60] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," 2019, *arXiv:1903.10082*.

[61] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, "Feedback network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3867–3876.

[62] T. Dai, J. Cai, Y. Zhang, S. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11065–11074.

[63] Y. Mei, Y. Fan, Y. Zhou, L. Huang, T. S. Huang, and H. Shi, "Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5689–5698.

[64] Y. Mei, Y. Fan, and Y. Zhou, "Image super-resolution with non-local sparse attention," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3517–3526.

[65] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 9992–10002.



Tingzhao Yu received the B.S. degree in automation from the Ocean University of China, Qingdao, China, in 2013, the Ph.D. degree in pattern recognition and intelligent systems from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2019. He is currently an Associate Professor with Public Meteorological Service Center, China Meteorological Administration, Beijing. His research interests include meteorological machine learning and large-scale spatial-temporal analysis.



Ruyi Yang received the B.S. degree in atmospheric science from the Nanjing University of Information Science and Technology, Nanjing, China, in 2014, the Ph.D. degree in mesoscale meteorology from the Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing, China, in 2019. She is currently an Associate Professor with Public Meteorological Service Center, China Meteorological Administration, Beijing. Her research interests include numerical modeling, orographic rainfall and meteorological machine learning.



Yan Huang received the Graduate degree majoring in atmospheric science from the Nanjing University of Information Science and Technology, Nanjing, China, in 2005, the M.S. degree majoring in applied meteorology from China Agricultural University, Beijing, China, in 2007, and the Ph.D. degree majoring in meteorology from the Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing, in 2010.

She is currently a Senior Engineer with Public Meteorological Service Center, China Meteorological Administration, Beijing. Her research interests include meteorological data fusion and assimilation, machine learning, and data quality control.



Jinbing Gao received the B.S. degree in computer science and technology from Shandong University, Jinan, China, in 2010, and the M.S. degree in computer science from Beihang University, Beijing, China, in 2013. He is currently an Associate Professor with Public Meteorological Service Center, China Meteorological Administration, Beijing. His research interests include meteorological machine learning and wind solar meteorological services.



Qiuming Kuang received the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2017. He joined the Public Weather Services Center, China Meteorological Administration (CMA), Beijing, in 2017. He is currently a Team Leader in AI and weather integration with Public Meteorological Service Center, CMA. His research interests include meteorological hazards forecasting and smart weather service.