

Super-Resolution Reconstruction of Humidity Fields based on Wasserstein Generative Adversarial Network with Gradient Penalty

Tao Li¹, Liang Wang^{2*}, Lina Wang¹ and Rui Han³

¹School of Artificial Intelligence, Nanjing University of Information Science & Technology, Nanjing, 210044, China

²School of Computer Science, Nanjing University of Information Science & Technology, Nanjing 210044, China

[e-mail: wangliang20200@163.com]

³Unit 93117 of PLA, PLA, Jiangsu, 210000, China

*Corresponding author: Liang Wang

*Received April 19, 2023; revised September 24, 2023; revised February 24, 2024;
accepted May 16, 2024; published May 31, 2024*

Abstract

Humidity is an important parameter in meteorology and is closely related to weather, human health, and the environment. Due to the limitations of the number of observation stations and other factors, humidity data are often not as good as expected, so high-resolution humidity fields are of great interest and have been the object of desire in the research field and industry. This study presents a novel super-resolution algorithm for humidity fields based on the Wasserstein generative adversarial network(WGAN) framework, with the objective of enhancing the resolution of low-resolution humidity field information. WGAN is a more stable generative adversarial networks(GANs) with Wasserstein metric, and to make the training more stable and simple, the gradient cropping is replaced with gradient penalty, and the network feature representation is improved by sub-pixel convolution, residual block combined with convolutional block attention module(CBAM) and other techniques. We evaluate the proposed algorithm using ERA5 relative humidity data with an hourly resolution of $0.25^{\circ} \times 0.25^{\circ}$. Experimental results demonstrate that our approach outperforms not only conventional interpolation techniques, but also the super-resolution generative adversarial network(SRGAN) algorithm.

Keywords: super-resolution, generative adversarial network, relative humidity, deep learning, gradient penalty.

1. Introduction

Relative humidity is an indicator that describes the amount of water contained in the air, it indicates the amount of water vapor in the air and the temperature at which the ratio of the maximum amount of water vapor that can be held. The higher the value of relative humidity, the more water is present in the air. In meteorology, relative humidity is a crucial indicator for measuring the moisture content of the air. When the relative humidity reaches a certain level, the water vapor in the air can condense into clouds and rain, leading to precipitation. As such, meteorological disasters are closely linked to relative humidity. For example, in events such as tropical cyclones and heavy rain, the relative humidity in the air is often very high. By monitoring changes in relative humidity, meteorologists can predict and detect these meteorological disasters. However, obtaining high-quality relative humidity grid data can be challenging, owing to limitations in the number and distribution of meteorological observation stations.

Artificial intelligence technology is advancing rapidly, with its applications expanding across various fields, including meteorology. The integration of neural networks enables more accurate weather predictions, enhancing forecasting precision through intelligent algorithms. Data mining and big data analysis can also be used to analyze meteorological data and improve our understanding of weather trends. In conclusion, the integration of artificial intelligence with meteorology can help us better understand weather and provide more accurate forecasts. Image super-resolution reconstruction is a classic technique in the field of computer vision [1]. The rapid development of convolutional neural networks has led to significant advancements in image super-resolution technology, which has found successful applications in various fields of computer vision and image processing, including medical and satellite image processing [2][3]. Super-resolution technology enhances the resolution of a low-resolution image, typically by estimating and filling in missing pixel values through grid densification. Downscaling techniques for meteorological data involve transforming low-resolution, large-scale global climate models into high-resolution, regional climate models with the aim of obtaining meteorological data with higher spatial resolution. The common downscaling methods include dynamic downscaling and statistical downscaling [4]. Using a grid of $0.5^\circ \times 0.5^\circ$ as an example, within a range of 0.25° resolution, this area encompasses 2×2 grid point data. After undergoing a 5-fold super-resolution process, the original 2×2 grid points are further subdivided within the same range to yield a resolution of 0.05° , resulting in a grid of 10×10 points. This process exemplifies downscaling, where grid point data for a specific region undergo refinement to acquire more detailed information within the same extent. It has similar scenarios with super-resolution and can draw on the relevant technologies of artificial intelligence in super-resolution for the field of meteorological downscaling.

GANs [5] have become a popular type of deep model for image super-resolution in recent years. Generative adversarial network(GAN) consists of two models: the generator and the discriminator. The generator is responsible for generating new images, while the discriminator's role is to distinguish between the generated images and real images. During GAN training, the generator and discriminator interact with each other, with the generator improving its ability to generate more realistic images, while the discriminator becomes more proficient in distinguishing between real and generated images. This article applies the idea of GAN used in image super-resolution to the generation of humidity field super-resolution products, and proposes a GAN-based algorithm for humidity field downsampling. To overcome the problem of unstable training and difficult convergence of GAN, this paper uses GAN with Wasserstein distance and replaces gradient clipping with gradient penalty [6]. To

evaluate the proposed algorithm's effectiveness, ERA5 hourly relative humidity data [7] is used for super-resolution experiments and result analysis. This research makes the following main contributions:

1. The low-resolution images are convolved with convolutional kernels of different sizes, which can capture more details compared to using a single kernel size. This approach can effectively recover the details of low-resolution images.
2. To raise the generalization performance of the model, residual structure [8] and batch normalization [9] are used in this paper. CBAM [10] is also incorporated into the residual structure, which improves not only the generalization ability but also the feature representation ability and accuracy of the model.
3. By incorporating a recurrent structure to fuse temporal information, the model performance is enhanced. Downscaling and super-resolution, while sharing similarities, are distinct. Utilizing super-resolution techniques to address meteorological challenges represents a forward-looking approach. Through the application of deep learning, we aim to ensure data accuracy while elevating the reliability and practical value of forecasts. This presents a viable pathway for future exploration in the field.

2. Related work

Super-resolution techniques can be broadly categorized into conventional interpolation, frequency domain-based, and deep learning-based methods. There are also some classical methods for downscaling in the meteorological field that are related to the work presented in this paper [11].

Lanczos interpolation is a template algorithm that works by weighted averaging of the surrounding pixels of an image to enlarge it. The Lanczos function is a finite field function used to smooth the image during interpolation, and it is suitable for preserving sharp edges in the image. Thus, it is often used for image enlargement or reduction in image processing. Bilinear interpolation is a 2D image enlargement algorithm that interpolates the four surrounding points of each pixel and uses two straight lines to approximate the color value of that point. Bilinear interpolation is effective in image enlargement, but it cannot guarantee sharp edges in the image. Therefore, it is usually used for image smoothing and denoising. Bicubic interpolation enlarges the image by using two cubic functions to interpolate the image, which performs well in preserving the smoothness of the image and ensuring its sharpness. Bicubic interpolation is commonly used in image enlargement, smoothing, and denoising. Fig. 1 shows a schematic of bilinear and bicubic interpolation. Experimental results show that although bilinear and bicubic interpolation perform well in terms of Peak signal-to-noise ratio(PSNR) and Structure Similarity Index Measure(SSIM), they tend to make the image too smooth locally due to the similarity of brightness values with neighboring pixels after interpolation. Hence, utilizing deep learning techniques, especially those based on generative adversarial networks, is a superior option for super-resolution.

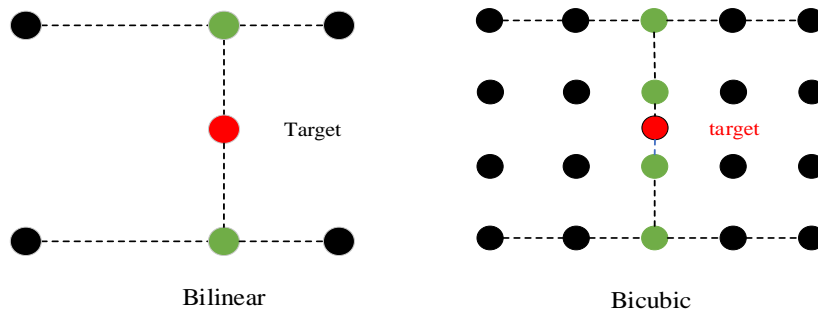


Fig. 1. Bilinear interpolation and bicubic interpolation

Super-resolution has seen significant advancements thanks to deep learning techniques. The Super-Resolution Convolutional Neural Network (SRCNN) [12] approach, based on convolutional neural network (CNN), was among the earliest methods developed for this purpose. Later, in terms of feature extraction, VSDR [13] allowed the network to predict the residual between the bicubic interpolation result and the real high-resolution image, which significantly improved the network performance compared to SRCNN. There are also some networks that use recursive structures, employing recursively connected convolutional layers or units. This approach gradually decomposes more challenging super-resolution problems into a set of simpler problems, making them easier to solve. Representative networks include DRCN [14], DRRN [15], and MenNet [16]. Among them, DRCN was the first network to apply recursive neural networks to image super-resolution. The introduction of generative adversarial networks revolutionized the field of super-resolution, and the SRGAN [17] technique utilizes the adversarial process between the generator and discriminator to produce high-resolution images that are visually perceptible. Fig. 2 illustrates the architecture of SRGAN. While SRGAN can generate more texture details, the textures often appear less natural and may come with some noise. Addressing the issues with SRGAN, Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) was introduced. ESRGAN [18] incorporates the Residual-in-Residual Dense Block (RRDB) and calculates perceptual loss using features before activation. In various tasks within the field of computer vision, including image super-resolution, convolutional neural networks (CNNs) play a crucial role. However, the emergence of attention-based networks, particularly those based on transformers [19], has revolutionized this landscape. The Swin Transformer [20] is a universal backbone network that has significantly outperformed previous records in tasks such as object detection and semantic segmentation. It has been widely applied in numerous visual tasks. SwinIR [21] is an image restoration model based on the Transformer architecture, which introduces the Transformer into image restoration tasks, achieving an end-to-end Transformer structure. Compared to existing models, SwinIR has fewer parameters and achieves superior results in various image restoration tasks, including denoising, super-resolution, and deblurring, setting state-of-the-art benchmarks.

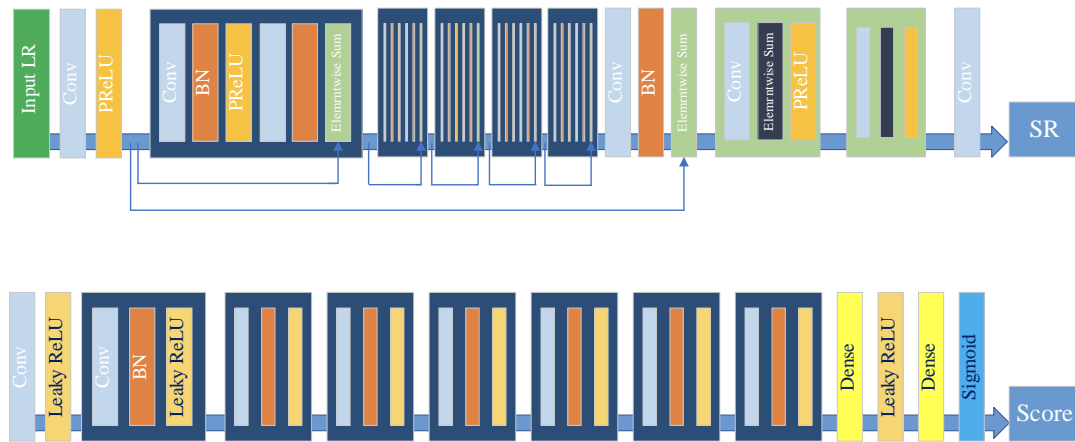


Fig. 2. SRGAN structure

The downscaling in meteorology has similarities with superresolution. Downscaling in meteorology is achieved by establishing a nonlinear mapping relationship between element fields at different scales. The goal of both is to convert low-resolution fields, or images with low image quality, into high-resolution fields, or images with high image quality. As deep learning continues to advance in the field of computer vision, an increasing number of researchers are exploring the use of deep learning-based super-resolution techniques for downscaling meteorological data. Leinonen et al. proposed a cyclic random super-resolution GAN to downscale atmospheric fields that evolve over time [22]. Vandal et al. introduced the first deep learning-based downscaling model DeepSD [23] for precipitation downscaling tasks by stacking multiple SRCNNs. Tie et al. introduced CLDASSD [24] to downscale the CLDAS 2m temperature product and generate a temperature product with higher resolution. Lucy Harris et al. introduced a GAN-based method [25] that can improve the resolution of precipitation forecasts by a factor of 10. Li Tao et al. used GANs to downscale the ERA5 2m temperature product and reconstruct a higher-resolution temperature product [26]. Gabriele Accarino et al. proposed a method called MSG-GAN-SD [27] based on Wasserstein Generative Adversarial Nets - Gradient Penalty (WGAN-GP). Through monthly and quarterly training as well as the use of different discriminator update strategies, they effectively performed statistical downscaling on 2m temperature fields. Tomoki Izumi et al. [28] used ESRGAN for super-resolution of sea surface temperatures. They evaluated and compared six subregions in three regions: Japan, North America, and South America. Their findings indicated that GANs can effectively represent complex flow field information, providing value for water resource management. Jinah Kim et al. proposed a multi-source data fusion and super-resolution method based on WGAN-GP [29]. They conducted experiments using data from global oceans and the surrounding waters of Korea, generating accurate downscaled sea surface temperature products. OPHÉLIA MIRALLES et al. [30] utilized WGAN-GP with ERA5 reanalysis data as low-resolution input and the Swiss Meteorological Service's COSMO-1 numerical weather prediction model as high-resolution targets. The experiments covered the entire Swiss region. The generated high-resolution wind field showed good visual results, effectively capturing mountainous terrain effects. However, there is room for further improvement, especially in complex mountainous areas. Ye Yang et al. [31] combined GANs with high-resolution China Meteorological Forcing Dataset (CMFD) temperature products and ground observation-based TP surface temperature dataset. They generated a 0.1° resolution 2m temperature dataset covering the Qinghai-Tibet Plateau from 1979 to 2020. This data

effectively captures the spatial distribution and trends of surface temperature in the region. The study demonstrates the potential and future prospects of artificial intelligence methods in expanding high-resolution meteorological datasets. When using meteorological data, it is important to consider that some of it may involve privacy issues. Furthermore, in anticipation of future model deployment and applications, careful consideration of data sharing policies is necessary to strike a balance between security and openness. Employing various encryption methods such as Attribute-Based Encryption [32], Attribute-Based Proxy Re-Encryption [33], and Revocable Attribute-Based Encryption [34] can ensure flexible data sharing while safeguarding data privacy. Only by appropriately addressing data privacy and sharing concerns can deep learning models trained on extensive meteorological data be widely applied in the public domain.

3. Datasets

The experiment used the ERA5 hourly dataset [7], which contains global meteorological data at an hourly level. The European Centre for Medium-Range Weather Forecasts (ECMWF) provides the ERA5 hourly dataset that spans from 1940 to present, and contains a variety of meteorological variables such as humidity, temperature, precipitation, wind speed, and more. In this study, relative humidity was used for super-resolution, and the dataset has a resolution of $0.25^\circ \times 0.25^\circ$.

This paper focuses on Beijing and its surroundings. **Table 1** summarizes the dataset. The original data spans from 00:00 to 23:00 daily. Cropped data is sampled every three hours, resulting in eight time steps. Spatially, data is uniformly downsampled at a stride of 8 for lower resolution.

Table 1. Overview of the Dataset

Dataset	Data size	Temporal span	Spatial span
Training Set	731(Day)*1*8*8	Jan 2019-Dec 2020	105.0°E~120.75°E 37.0°N ~ 52.75°N
Validation Set	366(Day)*1*8*8	randomly select from the Training set	105.0°E~120.75°E 37.0°N ~ 52.75°N
Test Set	304(Day)*1*8*8	Jan 2022-Oct 2022	105.0°E~120.75°E 37.0°N ~ 52.75°N

4. Methodology

4.1 Models

The experiment utilizes a GAN that comprises a generator and a discriminator. The generator learns the distribution of sample data and takes low-resolution meteorological fields as input to generate high-resolution meteorological fields that are as realistic as possible. The discriminator, on the other hand, is tasked with accurately determining whether the input meteorological fields are real or generated "fake data" by the generator. The following introduces the structure of the model and the sub-modules used, including Squeeze-and-Excitation network (SENet) [35], residual structure [8], CBAM [10], convolutional gate recurrent unit (ConvGRU) [36], and sub-pixel convolution [37].

The proposed humidity field super-resolution model network structure in this paper is shown in Fig. 3. The data with the shape of $D \times T \times 1 \times H \times W$ is input into the generator, and different sizes of convolution kernels (2×2 , 3×3) are used to expand the channels to 128. The resulting output is then added with Gaussian noise to improve the model's generalization performance. After that, the input data is processed by the SENet block for channel attention, which increases the weight of important channels. The data then goes through multiple residual blocks, where CBAM is added to the end of each residual block to improve feature extraction capabilities. Using residual structures can also solve the problem of gradient disappearance during training as network depth increases. Different sizes of convolution kernels are used in these convolutional layers to extract different features, fully utilizing the information in the image and effectively improving the image detail restoration effect.

After processing through the residual blocks, the network employs ConvGRU units. ConvGRU units combine the benefits of convolutional and recurrent neural networks by performing convolutions at each time step to preserve spatial correlations of the convolutional features, while using the recurrent unit to learn temporal dependencies. The output of the ConvGRU layer is then expanded to 256 channels through several residual blocks and finally upsampled using sub-pixel convolution to double the size of the image at each upsampling step. In the final layer, a 1×1 convolutional kernel is used to map the output to the expected feature map shape.

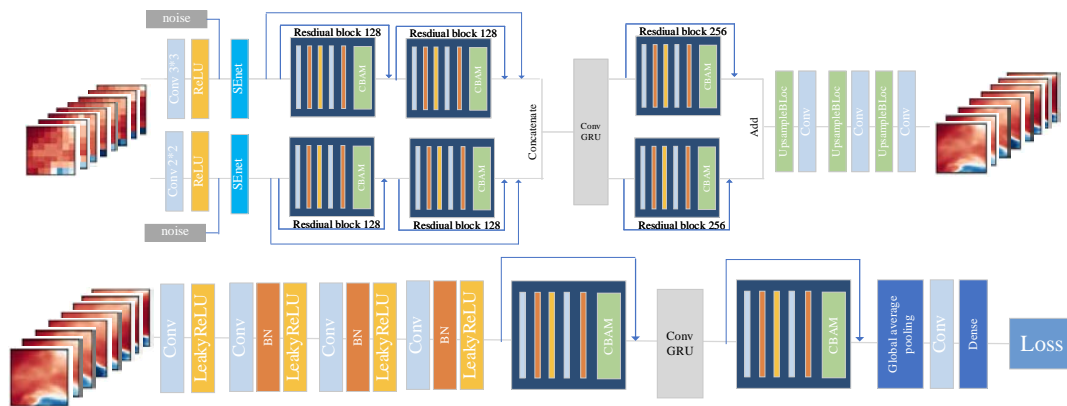


Fig. 3. Generator and discriminator structure

The discriminator discriminates the input images. The image is first passed through convolution and several residual blocks, followed by ConvGRU and more residual blocks. Finally, global average pooling is performed on each feature map of the input, computing the global average value and retaining it. This compresses each feature map into a single value, ensuring that the network is insensitive to the input image size. Furthermore, the reduction in the number of features also reduces the number of network parameters by performing global average pooling. The final layer of the network is a fully connected layer that produces a one-dimensional vector indicating the difference in probability density between the input image's distribution and the distributions of the generated images and real images. The WGAN-GP model employs the Wasserstein distance to measure the divergence between the distributions of the generated and real images. The discriminator's output reflects this divergence.

The main function of convolution operation is to expand the receptive field of the neural network and fuse more features spatially. However, in the channel dimension, traditional convolution operations usually merge all input feature maps equally, which is unreasonable in practice. This is because different channel features contribute differently to the output, with some channels containing critical information and others being less important. Therefore, to more effectively utilize the information between channels, it is necessary to fuse features for each channel and assign different weights to capture critical information and improve network performance and generalization ability. Jie Hu et al.'s SENet focuses well on the relationship between channel dimensions, and Squeeze and Excitation are two crucial operations designed to explicitly model the interrelationship between feature channels. Fig. 4 shows the structure of SENet. The Squeeze operation first performs global average pooling on each channel to reduce the features of each channel to a scalar value, compressing the shape of the feature map from $C \times H \times W$ to $C \times 1 \times 1$, essentially transforming the original two-dimensional feature channel into one dimension, allowing this dimension to acquire a global view of the previous $H \times W$. This is shown in (1). Excitation maps the global features to a channel weight vector through two fully connected layers for weighting the channel features. This is shown in (2).

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W u_c(i, j) \quad (1)$$

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2(\text{ReLU}(W_1 z))) \quad (2)$$

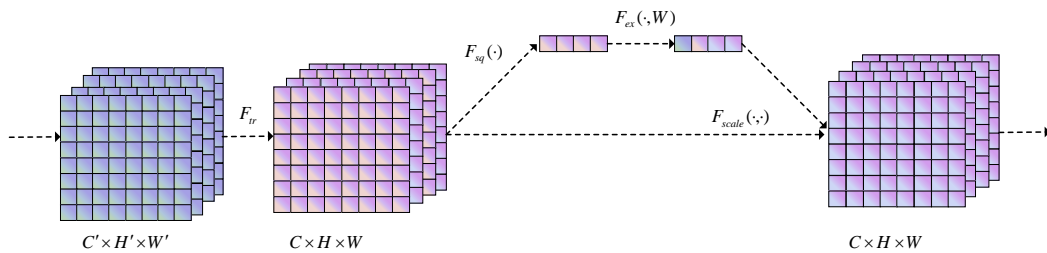


Fig. 4. SENet structure

Fig. 5(a) shows the structure of a residual block. The residual block uses convolution, batch normalization, and ReLU activation function. Stacking residual blocks together to form a residual structure can alleviate the problems of gradient vanishing and insufficient network expression ability. Additionally, the simple structure of residual structures contributes to the network's stability and ease of training. At the end of the residual block is a CBAM. CBAM is an attention mechanism used in convolutional neural networks to improve the quality of feature representation. It mainly consists of two parts: CAM and SAM. CAM can learn the importance weights of each channel, so as to better capture useful information in each channel and make the CNN focus on the important features of each channel without being disturbed by irrelevant features. SAM can learn to weight specific regions in space to better focus on important spatial features and ignore unimportant regions. CBAM can be conveniently embedded into different levels of CNN and be used in existing network architectures. Fig. 5(b) illustrates the schematic diagram of sub-pixel convolution, and To increase the resolution of a feature map by a factor of 2, a convolutional operation is first used to expand the number of channels by a factor of 4 to $(1 \times 2^2, H, W)$. Finally, the number of channels is reduced to 1 for pixel recombination. The resulting size will be $2H \times 2W$.

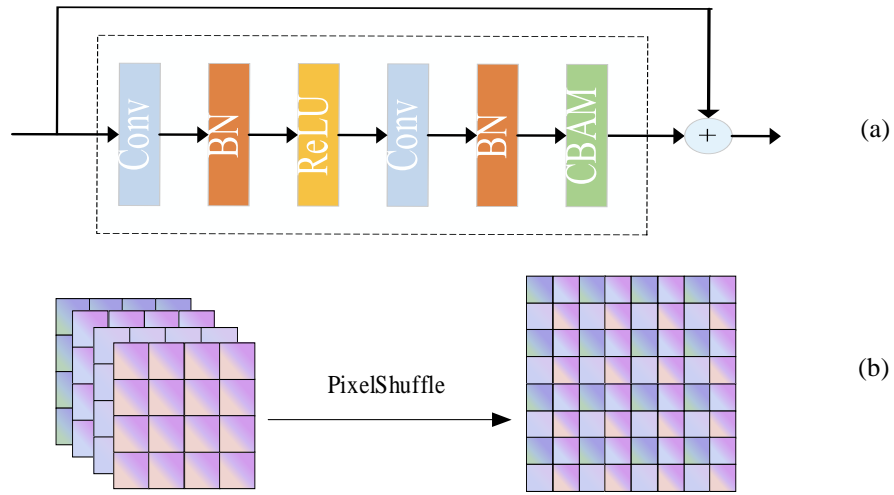


Fig. 5. Residual blocks and PixelShuffle

The CAM structure is shown in Fig. 6. First, the feature map F with a shape of $H \times W \times 1$ is processed through global average pooling and global max pooling to obtain two $1 \times 1 \times C$ feature maps. Then, they are fed into a shared two-layer neural network composed of multiple perceptrons and a hidden layer. Finally, the features output by the neural network are added together, and then the sigmoid activation operation is applied to limit the weights in $[0, 1]$, resulting in the required channel attention feature. Multiplying the channel attention feature with the input feature map F produces the weighted channel feature map. The weighted design approach is simple and efficient, directly enhancing or suppressing channels in the input feature maps. This not only avoids a significant increase in parameters and computations but also makes the network structure modular and easy to integrate into different convolutional networks. Therefore, the channel attention mechanism can make the network more focused on important features of each channel, improving the network's capability, as shown in (3).

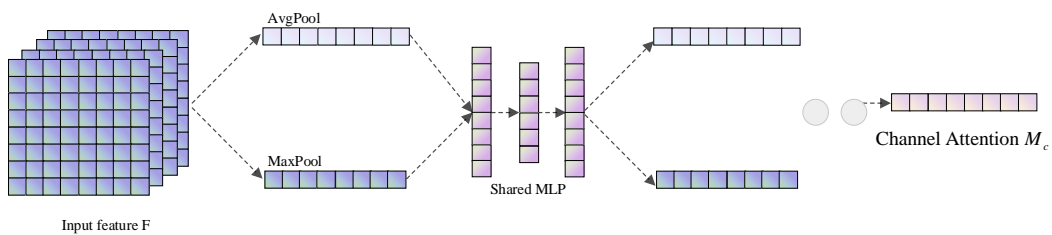


Fig. 6. CAM structure

$$\begin{aligned}
 M_c(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\
 &= \sigma(W_1(W_0(F_{avg}^C)) + W_1(W_0(F_{max}^C)))
 \end{aligned}
 \tag{3}$$

The SAM structure is shown in Fig. 7. Similar to the channel attention mechanism, the feature map is first passed through global average pooling and global maximum pooling to obtain two $H \times W \times 1$ feature maps. Then, these two feature maps are combined along the channel dimension and fed into a 7×7 convolutional layer to reduce the number of channels to 1, resulting in an output feature map of size $H \times W \times 1$. Finally, the sigmoid activation is applied to obtain the spatial attention feature. Multiplying this feature map with the input feature map F results in the weighted spatial feature map. This is shown in (4).

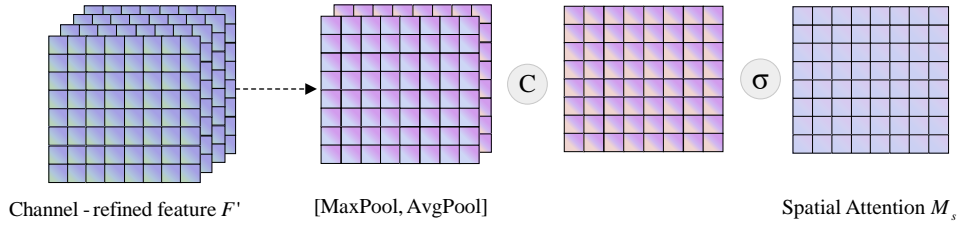


Fig. 7. SAM structure

$$\begin{aligned} M_s(F) &= \sigma(f^{7 \times 7}([\text{AvgPool}(F); \text{MaxPool}(F)])) \\ &= \sigma(f^{7 \times 7}([F_{avg}^S; F_{max}^S])) \end{aligned} \quad (4)$$

4.2 Model optimization objectives

The optimization objective of the generative adversarial network is to make the generated samples by the generator approach the real data distribution as much as possible, while making the discriminator unable to distinguish between real and generated data. This is achieved by using the adversarial nature of the generator and discriminator, usually by maximizing the log-likelihood for the generator and maximizing the discrimination accuracy for the discriminator. The objective function is shown in (5), where x represents the real high-resolution image, z represents the input low-resolution image, and $G(Z)$ represents the super-resolution image generated by the generator.

$$\min_G \max_D V(D, G) E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (5)$$

The discriminator evaluates either the real high-resolution image or the generator's super-resolution output, producing a real number between 0 and 1 to indicate the authenticity of the input sample. The closer the output is to 1, the more the discriminator thinks the input data is closer to the real sample, and the closer it is to 0, the more the discriminator thinks the input data is closer to the fake sample. Taking the \log of the output of the discriminator D , such as $\log D(x)$ and $\log(1 - D(G(z)))$, is a common way to construct the loss function of the discriminator model. The role of logarithm is to map the numbers in the $[0, 1]$ interval to the range $(-\infty, 0]$, so that it can be derived and optimized by gradient descent. The generator G takes the low-resolution image as input and generates the super-resolution image. Its objective is exactly the opposite of the discriminator's, that is, to make the discriminator identify its generated samples (super-resolution images) as real samples. $E_{x \sim p_{data}(x)}[\log D(x)]$ represents the expected value of the discriminator's judgment on the real samples. For the optimal discriminator D^* , the judgment result $D(x)$ on the real sample is 1, and $\log D(x)$ is 0.

If the discriminator is not optimal, $\log D(x)$ is less than 0. In other words, if we want the discriminator to reach the optimal, $E_{x \sim P_{data(x)}}[\log D(x)]$ should be as large as possible, whereas for the generator, $E_{x \sim P_{data(x)}}[\log D(x)]$ should be as small as possible. Similarly, $E_{z \sim P_z(z)}[\log(1 - D(G(z)))]$ represents the expected judgment result of the discriminator for fake samples. For the optimal discriminator D^* , the judgment result for fake samples $D(G(z))$ should be 0, $1 - D(G(z))$ should be 1, and $\log(1 - D(G(z)))$ should be 0. If the discriminator is not optimal, $\log(1 - D(G(z)))$ is less than 0. In other words, if we want the discriminator to reach the optimal, $E_{z \sim P_z(z)}[\log(1 - D(G(z)))]$ should be as large as possible, and for the generator, $E_{z \sim P_z(z)}[\log(1 - D(G(z)))]$ should be as small as possible.

The optimization goal of GAN is a minimax problem. We aim to find the optimal discriminator D^* that maximizes the objective function $V(D, G)$. Furthermore, we want to continue searching for the optimal generator G^* that minimizes the objective function under the optimal discriminator, i.e., the generated samples make the discriminator perform as poorly as possible. The game theory idea of GAN is embodied here.

As the goal of GAN is to minimize the JS divergence between the generated samples and the real samples, in order to make the distribution of generated samples similar to that of real samples. However, when using JS divergence as the loss function for training, the gradient calculation involves the difference between the two distributions. When these two distributions are significantly different, the gradient of JS divergence can be very large, which may lead to the problem of gradient explosion. On the other hand, when these two distributions are very similar, the gradient of JS divergence becomes very small, which may cause the problem of gradient vanishing. WGAN [38], as an improved version of GAN, uses the Wasserstein distance (also known as the Earth-Mover (EM) distance) as the loss function instead of JS divergence. The approximation of the Wasserstein distance is achieved by restricting the discriminator's parameters, which is usually implemented through gradient clipping to limit the discriminator's output range to $[-c, c]$. However, gradient clipping limits the range of network parameter values, which may push the weights to two extreme values of the clipping range, causing the gradient to increase or decrease exponentially as it moves through the network. Therefore, the problem of gradient vanishing or explosion still exists [39][40][41]. To address this issue, the WGAN-GP method introduces gradient penalty in the discriminator. Equation (6) shows the objective function of the WGAN-GP. During the training process, we adopted the penalty coefficient suggested in the original WGAN-GP paper and set λ to 10.

$$\min_G \max_D E_{z \sim P_z(z)} [D(G(z))] - E_{x \sim P_{data}(x)} [D(x)] + \lambda E_{\hat{x} \sim P_{\hat{x}}} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \quad (6)$$

The L2 loss is replaced by the L1 loss in the generator, and in a growing number of studies it has also been shown that using the L1 loss gives better experimental results [42]. Firstly, L1 loss is more robust than L2 loss, which means it penalises outliers less. Second, L2 loss leads to more smoothing, making the results more blurred.

During training, the learning rate is updated using a cosine annealing decay strategy. The main idea of cosine annealing is to gradually reduce the learning rate during training in order to avoid convergence problems caused by too large a learning rate. Usually, when the learning rate is large, the model will approach the optimal solution quickly, but if the learning rate is too large, it will overshoot and cross the optimal solution. In this case, the learning rate needs to be gradually reduced so that the model converges near the optimal solution. The update strategy is shown in (7). η_{min} denotes the minimum learning rate and the default is 0. lr_0 represents the initial learning rate. $epoch_i$ denotes the current number of iterations.

T_{max} represents the maximum number of iterations required for one cosine cycle. LR_{sch} denotes the standard cosine annealing strategy. warmUpLR denotes the warm-up learning rate strategy. In the beginning of training, when the current number of iterations is less than the set warm_epoch, WarmUpLR is used. Since the model's weights are randomly initialized at the beginning of training, choosing a large learning rate may cause instability. To improve training stability and accelerate model convergence, a warm-up learning rate is used where the initial few epochs have a smaller learning rate.

$$LR_{sch} = (epoch_i, T_{max}, eta_{min}) = eta_{min} + 0.5 \times (lr_0 - eta_{min}) \times (1 + \cos \frac{epoch}{T_{max}}) \quad (7)$$

$$WarmUpLR = base_lr \times batch_size / (total_iters + 1e - 8)$$

The training process of a neural network generally includes the following steps.

Algorithm 1: Network Training Process

Input: training set Data, integer epochs, integer warm_epoch, gradient penalty coefficient

λ , warm_sch, lr_sch

Output: Reconstructed SR images

for t ← 1 to epochs do

 for k ← 1 to K do

 1. Sample m low-resolution samples from training set Data

$$\{z^1, z^2, z^3 \dots, z^m\}.$$

 2. Sample m high-resolution samples from training set Data

$$\{x^1, x^2, x^3 \dots, x^m\}.$$

 3. Sample a random number $\varepsilon \sim U[0,1]$

 4. Update the discriminator parameters.

$$\hat{x} \leftarrow \varepsilon x^i + (1 - \varepsilon) G(z^i)$$

$$L_D^i \leftarrow D(G(z^i)) - D(x^i) + \lambda (\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2$$

$$w \leftarrow Adam(\nabla_w \frac{1}{m} \sum_{i=1}^m L_D^i, w, \alpha, \beta)$$

 5. Update the generator parameters.

$$\theta \leftarrow Adam(\nabla_{\theta} \frac{1}{m} \sum_{i=1}^m (-D_w(G_{\theta}(z^i))), \theta, \alpha, \beta)$$

 6. Update the learning rate.

$$\text{if } t \geq \text{warm_epoch}: lr \leftarrow lr_sch(t, lr_0)$$

$$\text{else}: lr \leftarrow \text{warm_sch}(\text{warm_epoch} * \text{len}(\text{Data}))$$

 end for

end for

4.3 Evaluation method

To better evaluate the model's performance, this paper employs metrics including Mean Squared Error(MSE), Mean Absolute Error(MAE), PSNR, SSIM, Anomaly Correlation Coefficient(ACC), and Temporal Correlation Coefficient(TCC) to assess the quality of

generated images. Among these, MSE, MAE, PSNR, and SSIM are widely used in the field of computer vision, while ACC and TCC are commonly used metrics in meteorological applications. MSE is a statistical indicator used in statistical learning and machine learning to measure the deviation of data from the true value. The formula for MSE is shown in (8).

$$MSE = \frac{1}{N} \sum_{i=1}^N (G_i - X_{real,i})^2 \quad (8)$$

Where N is the number of pixels, G_i is the generated image, and $X_{real,i}$ is the reference real image. MSE represents the average squared difference between the predicted and true values, and the smaller the MSE, the closer the generated image is to the real image. However, MSE mainly focuses on the overall image error and may not be sensitive enough to errors in the details.

MAE is a commonly used error metric in machine learning. It is based on the L_1 norm and measures the average absolute difference between predicted and actual values. The formula for MAE is shown in (9).

$$MAE = \frac{1}{N} \sum_{i=1}^N |G_i - X_{real,i}| \quad (9)$$

PSNR is an image quality metric that compares the peak power of an image to its noise level. The formula for PSNR is shown in (10).

$$PSNR = 10 \log_{10} \frac{I_{max}^2}{MSE} \quad (10)$$

PSNR is a widely used image quality metric in various fields of image processing, including image compression and restoration. Generally, The quality of the image improves as the PSNR value increases. One notable feature of PSNR is that it is not sensitive to the relative size of the images, making it very simple and easy to use. However, the accuracy of PSNR in handling high-contrast images needs improvement.

SSIM [43] is used to evaluate the similarity between two images. The SSIM values are between 0 and 1, where a value of 1 indicates that the two images are identical, and a value of 0 indicates that they are completely dissimilar. It is a similarity metric based on structure, contrast, and brightness, rather than simply evaluating pixel differences. Compared with PSNR and MSE, SSIM considers factors such as structure, contrast, and brightness, which are more similar to human perception of image quality, making it more suitable for evaluating image processing tasks. The calculation of SSIM is shown in (11).

$$\begin{aligned} L(G, X) &= \frac{2\mu_G\mu_X + C_1}{\mu_G^2 + \mu_X^2 + C_1} \\ C(G, X) &= \frac{2\sigma_G\sigma_X + C_2}{\sigma_G^2 + \sigma_X^2 + C_2} \\ S(G, X) &= \frac{\sigma_{GX} + C_3}{\sigma_G\sigma_X + C_3} \\ SSIM &= L(G, X) \times C(G, X) \times S(G, X) \end{aligned} \quad (11)$$

The ACC primarily indicates the spatial compatibility, also known as spatial similarity, between the predicted values (i.e., the results obtained from the model) and the ground truth. The calculation formula is as follows.

$$ACC = \frac{\frac{1}{N} \sum_{i=1}^N (f_i - \bar{f})(o_i - \bar{o})}{\sqrt{\frac{1}{N} \sum_{i=1}^N (f_i - \bar{f})^2 \frac{1}{N} \sum_{i=1}^N (o_i - \bar{o})^2}} \quad (12)$$

In this formula, N represents the total number of grid points in the study area, where f_i and o_i respectively denote the predicted value and the actual observation of the i th spatial grid point at a specific time point. \bar{f} and \bar{o} represent the means of the predicted values and actual observations, respectively, calculated over eight time steps in a day. The final ACC is calculated on a daily basis, averaging the ACC over eight time steps to obtain the daily ACC.

The TCC can statistically demonstrate the climate model's or model's ability to predict anomalies at various grid points. TCC is typically calculated using climate anomaly percentages. The range of TCC is between -1 and 1, with values closer to 1 indicating higher predictive skill. The calculation formula is as follows.

$$TCC = \frac{\frac{1}{T} \sum_{t=1}^T (f_t - \bar{f})(o_t - \bar{o})}{\sqrt{\frac{1}{T} \sum_{t=1}^T (f_t - \bar{f})^2 \frac{1}{N} \sum_{t=1}^T (o_t - \bar{o})^2}} \quad (13)$$

In this formula, T represents the length of time in the study area, where T is set to 8 for eight time steps in a day. f_t and o_t respectively denote the predicted value and the actual observation of the t th time point for a specific spatial grid point. \bar{f} and \bar{o} represent the means of T predicted values and actual observations, respectively.

5. Experimental results

During training, the Adam optimizer was used, and the Two Time-Scale Update Rule [44] strategy was adopted to achieve faster and more stable convergence. This means that different learning rates were used for the generator and discriminator during training because they have different objectives and loss functions. Using the same learning rate may result in an imbalance between the two. Therefore, the learning rates of the generator and discriminator are set to 0.0001 and 0.001, respectively, during training. The various indicators during training are shown in Fig. 8.

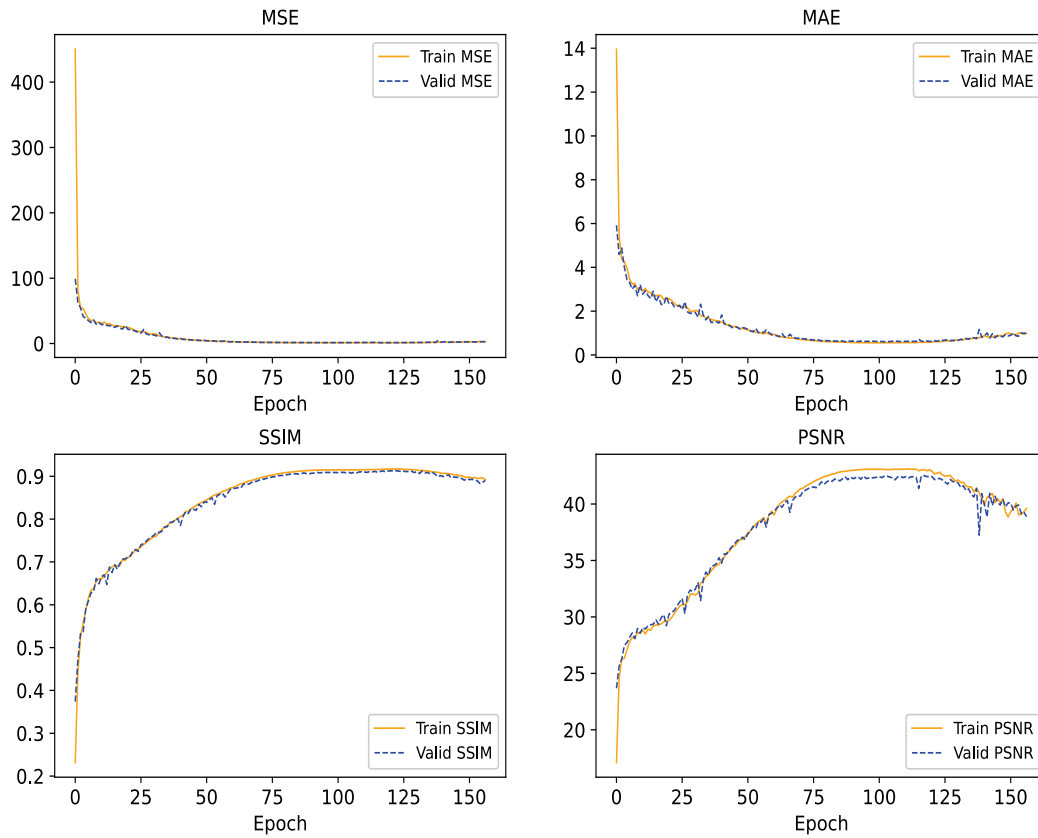


Fig. 8. Performance of each index during training

The model in this paper utilizes CBAM in conjunction with residual network(ResNet) modules, along with a Wasserstein objective function incorporating gradient penalty, abbreviated as the WGAN-GP module. In order to comprehend the roles these modules play in the model, aiding in the identification of crucial components and evaluating their contributions, as well as providing robust guidance for future improvements and optimizations, several ablation experiments were conducted, and the results are shown in **Table 2**.

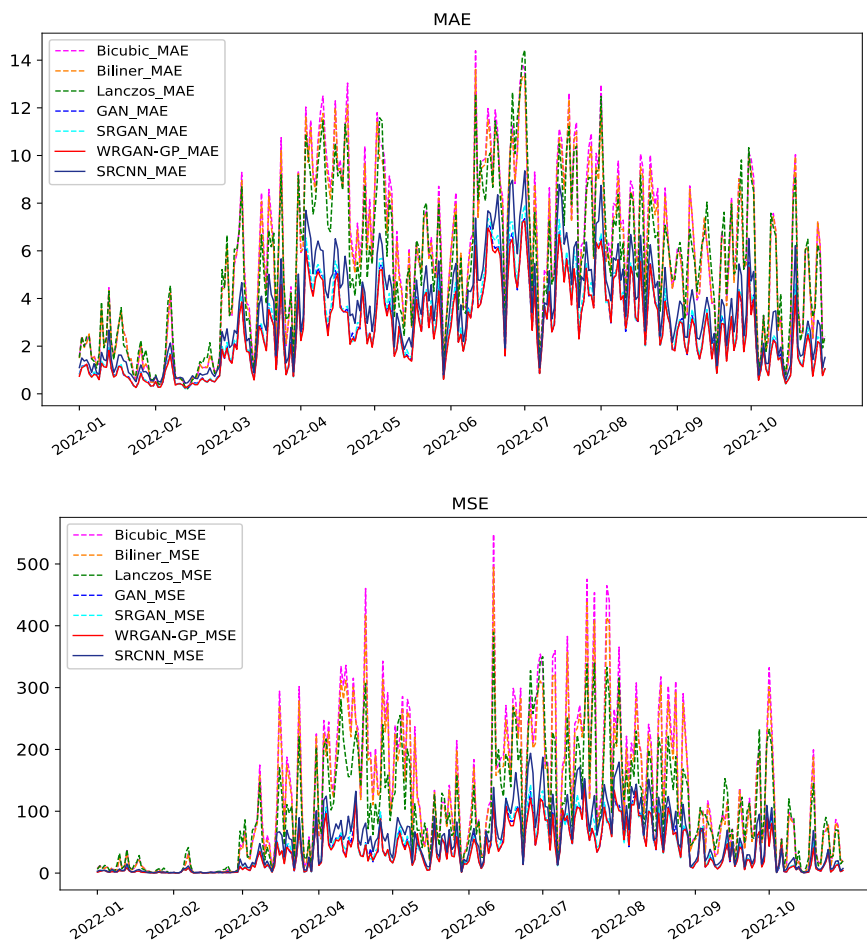
Table 2. Ablation Experiment Results

CBAM	ResNet	WGAN-GP	PSNR	SSIM	ACC	TCC
✓		✓	25.747	0.537	0.866	0.692
✓	✓		31.640	0.752	0.948	0.860
	✓	✓	31.483	0.753	0.948	0.861
✓	✓	✓	31.832	0.766	0.951	0.866

From the first row of the table, it can be seen that the performance of various metrics is significantly poor when ResNet is not utilized. Moreover, during the experiments, the model's convergence speed was much slower compared to the case where ResNet was employed. The last row demonstrates a notable improvement in metric performance with the use of ResNet,

which also reflects the effectiveness of ResNet. From the second and last rows, it can be observed that employing Wasserstein with gradient penalty as the optimization objective for the discriminator leads to an enhancement in metrics. The adoption of this new optimization objective has a positive impact on the model, which is one of the reasons why WGAN-GP was used in this paper. On the other hand, incorporating CBAM into the model's architecture improves the quality of feature representation. This can be discerned from the results of the third and last rows. Therefore, considering the performance of these modules collectively, utilizing the current architecture is a preferable choice.

We evaluated several interpolation methods, including bilinear, bicubic, Lanczos, and our proposed model, on the test set, which contained data from the first 304 days of 2022. Fig. 9 shows the results of this evaluation. Regardless of whether the model in this study was trained using a standard GAN or the WGAN-GP method, its performance metrics were superior to interpolation and SRGAN. From a seasonal perspective, for the months of January to March, both SSIM and PSNR metrics showed that GAN-based methods outperformed interpolation-based methods. Although this gap narrowed over time, there was still a noticeable advantage.



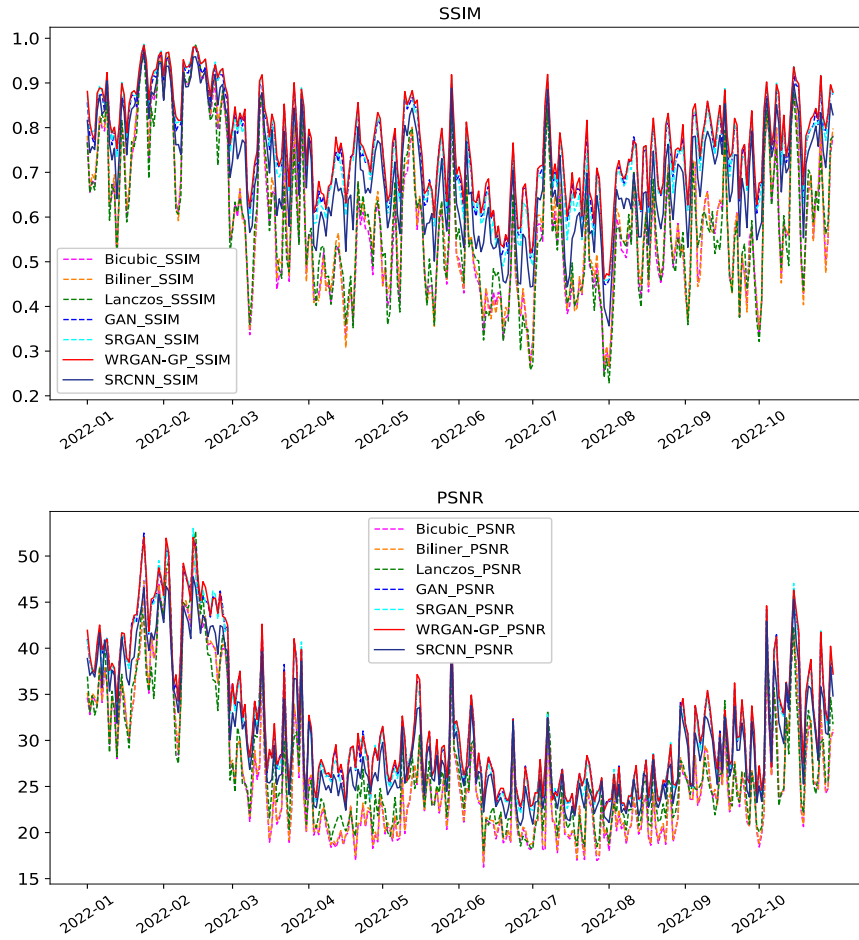


Fig. 9. Performance of each indicator on the test set

The average values of various indicators for different models and methods on the test set are listed in **Table 3**. Our proposed model shows improvements in both SSIM and PSNR as compared to traditional interpolation methods, indicating the potential of the model. The performance is also better than SRCNN and GAN-based SRGAN.

Table 3. Metrics performance of different methods on the test set

	MSE	MAE	SSIM	PSNR	ACC	TCC
Lanczos	99.889	5.432	0.593	27.394	0.890	0.741
Bilinear	114.449	5.592	0.601	26.669	0.879	0.747
Bicubic	122.472	5.762	0.593	26.418	0.871	0.728
SRCNN	50.143	3.499	0.699	29.973	0.904	0.819
GAN	37.362	2.755	0.752	31.640	0.948	0.860
SRGAN	38.611	2.863	0.749	31.494	0.948	0.848
WRGAN-GP	36.185	2.698	0.766	31.832	0.951	0.866

The effectiveness of methods based on GAN networks is evidently superior to that of interpolation methods. Comparing the results obtained on the same test set with those of SRGAN, this study's model shows an improvement in performance in terms of MSE, MAE, SSIM, and PSNR. The effectiveness of the proposed model in this paper has been verified. Some examples randomly selected from the test set are presented in [Fig. 10](#).

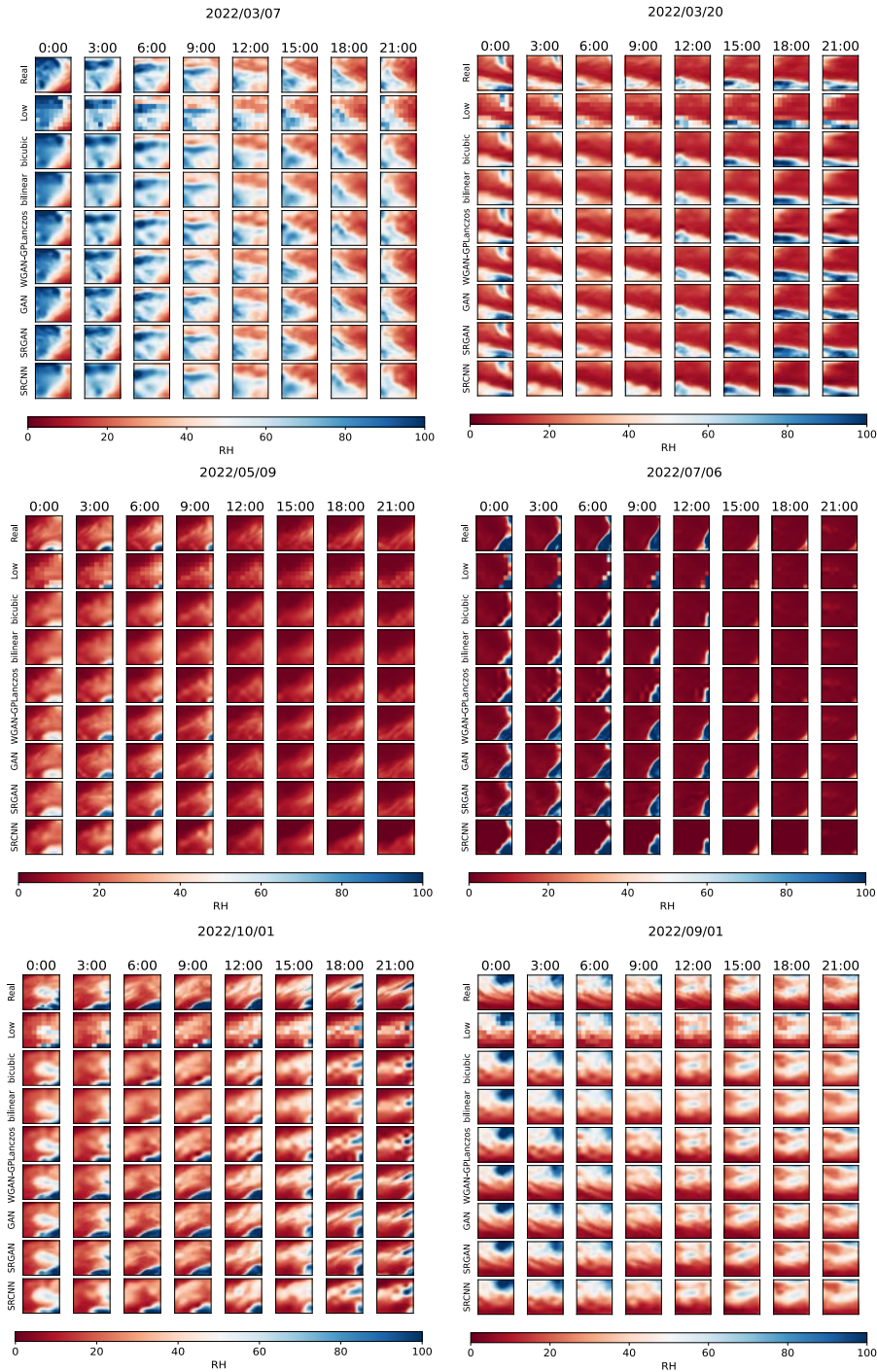


Fig. 10. Sample Test Set

6. Conclusion

When comparing models based on GANs with traditional interpolation methods, experimental results show that GAN-based models perform better. The model proposed in this paper also outperforms conventional GAN models, but due to the limitations of GAN networks, this method still has some shortcomings. For example, training difficulties, model instability, and the tendency of GAN models to fall into local optima can lead to poor image quality. Although these issues did not appear in the experiments, they still need to be addressed, and there is still room for improvement in the proposed model and work.

This study is evaluated using the ERA5 dataset, chosen for its high-quality reanalysis data provided by the European Centre for ECMWF. ERA5 boasts superior spatiotemporal resolution and relies on reliable observational data sources. This makes ERA5 an ideal choice for researching and analyzing meteorological, climatic, and related environmental issues. Many cutting-edge studies and assessments are built upon the foundation of ERA5, and obtaining similarly dependable datasets can be challenging. In the future, there may be opportunities to assess our model on a broader range of datasets. Even though the model in this paper outperforms SRAGN in terms of performance, there is no advantage in training and inference speed. Therefore, it needs to continue to optimize the network structure. The current model in this paper can only zoom in 8 times, which is not flexible enough. In the future, we hope to be more flexible to generate high-resolution images with multiple magnification ratios. The present work only has a single meteorological element, and it is hoped that more meteorological elements can be incorporated to further generate more realistic high-resolution images. The current super-resolution is performed spatially, and in the subsequent optimization is extrapolated temporally to achieve short-term forecasts. Downscaling and super-resolution share similarities but are not entirely identical. The downscaling issue for humidity, compared to other elements like temperature, is more intricate. While super-resolution methods can yield certain results, they may potentially violate physical principles or overlook inter-variable dependencies. In the future, it is hoped that incorporating physical constraints and integrating domain expertise into the model will lead to more rational outcomes. Deep learning models have tremendous potential in weather forecasting and environmental monitoring, and can be deployed in public agencies like weather bureaus. However, data sharing and privacy issues must be carefully considered first. Compute resources and maintenance costs are also critical. In summary, this is a valuable topic with proper research, better meteorological services and decision support can be achieved.

References

- [1] M. Irani and S. Peleg J. S. Turner, "Improving resolution by image registration," *CVGIP: Graphical models and image processing*, vol. 53, no. 3, pp. 231-239, May. 1991. [Article \(CrossRefLink\)](#)
- [2] M. Havaei, et al, "Brain tumor segmentation with deep neural networks," *Med. Image Anal.*, vol. 35, pp. 18-31, Jan. 2017. [Article \(CrossRefLink\)](#)
- [3] M. Pritt and G. Chern, "Satellite image classification with deep learning," in *Proc. of 2017 IEEE applied imagery pattern recognition workshop (AIPR)*, IEEE, pp. 1-7, Sept. 2017. [Article \(CrossRefLink\)](#)
- [4] R.L. Wilby and T. Wigley, "Precipitation predictors for downscaling: observed and general circulation model relationships," *International Journal of Climatology: A Journal of the Royal Meteorological Society*, vol. 20, no. 6, pp. 641-661, May. 2000. [Article \(CrossRefLink\)](#)
- [5] I. Goodfellow, et al, "Generative Adversarial Networks," *Commun. Acn*, vol. 63, no. 11, pp. 139-144, Oct. 2020. [Article \(CrossRefLink\)](#)

- [6] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A.C. Courville, "Improved training of wasserstein gans," *arXiv:1704.00028*, Mar. 2017. [Article \(CrossRefLink\)](#)
- [7] B. Bell, et al, "The ERA5 global reanalysis: Preliminary extension to 1950," *Q. J. R. Meteorol. Soc.*, vol. 147, no. 741, pp. 4186-4227, Sept. 2021. [Article \(CrossRefLink\)](#)
- [8] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE conference on computer vision and pattern recognition(CVPR)*, pp. 770-778, Dec. 2016. [Article \(CrossRefLink\)](#)
- [9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. of International conference on machine learning(ICML)*, pp. 448-456, 2015. [Article \(CrossRefLink\)](#)
- [10] S. Woo, J. Park, J. Lee and I.S. Kweon, "Cbam: Convolutional block attention module," in *Proc. of the European conference on computer vision (ECCV)*, pp. 3-19, Oct. 2018. [Article \(CrossRefLink\)](#)
- [11] B.C. Hewitson and R.G. Crane, "Climate downscaling: techniques and application," *Clim. Res.*, vol. 7, no. 2, pp. 85-95, Nov. 1996. [Article \(CrossRefLink\)](#)
- [12] C. Dong, C.C. Loy, K. He and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295-307, Feb. 2016. [Article \(CrossRefLink\)](#)
- [13] J. Kim, J.K. Lee and K.M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 1646-1654, Dec. 2016. [Article \(CrossRefLink\)](#)
- [14] J. Kim, J.K. Lee and K.M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 1637-1645, Dec. 2016. [Article \(CrossRefLink\)](#)
- [15] Y. Tai, J. Yang and X. Liu, "Image super-resolution via deep recursive residual network," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 3147-3155, Nov. 2017. [Article \(CrossRefLink\)](#)
- [16] Y. Tai, J. Yang, X. Liu and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proc. of 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4549-4557, Dec. 2017. [Article \(CrossRefLink\)](#)
- [17] C. Ledig, et al, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 105-114, Nov. 2017. [Article \(CrossRefLink\)](#)
- [18] X. Wang, et al, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proc. of ECCV 2018 workshop, arXiv:2005.00765*, Sept. 2018. [Article \(CrossRefLink\)](#)
- [19] A. Vaswani, et al, "Attention is all you need," *arXiv:1706.03762*, Aug.2023. [Article \(CrossRefLink\)](#)
- [20] Z. Liu, et al, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. of the IEEE/CVF international conference on computer vision*, pp. 9992-10002, 2021. [Article \(CrossRefLink\)](#)
- [21] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool and R. Timofte, "Swinir: Image restoration using swin transformer," in *Proc. of the IEEE/CVF international conference on computer vision Workshops*, pp. 1833-1844, Sep. 2021. [Article \(CrossRefLink\)](#)
- [22] J. LeinonenD. Nerini and A. Berne, "Stochastic super-resolution for downscaling time-evolving atmospheric fields with a generative adversarial network," *IEEE Trans. Geosci. Remote Sensing*, vol. 59, no. 9, pp. 7211-7223, Sep. 2021. [Article \(CrossRefLink\)](#)
- [23] T. Vandal, E. Kodra, S. Ganguly, A. Michaelis, R. Nemani and A.R. Ganguly, "DeepSD: Generating high resolution climate change projections through single image super-resolution," in *Proc. of the 23rd acm sigkdd international conference on knowledge discovery and data mining(KDD)*, pp. 1663-1672, Aug. 2017. [Article \(CrossRefLink\)](#)
- [24] R. Tie, C. Shi, G. Wan, X. Hu, L. Kang and L. Ge, "CLDASSD: Reconstructing Fine Textures of the Temperature Field Using Super-Resolution Technology," *Adv. Atmos. Sci.*, vol. 39, no. 1, pp. 117-130, Jan. 2022. [Article \(CrossRefLink\)](#)

- [25] L. Harris, A.T. McRae, M. Chantry, P.D. Dueben and T.N. Palmer, "A generative deep learning approach to stochastic downscaling of precipitation forecasts," *J. Adv. Model. Earth Syst.*, vol. 14, no. 10, pp. e2022M-e3120M, Oct. 2022. [Article \(CrossRefLink\)](#)
- [26] T. Li, Z. Jiang, R. Han, J. Xia and Y. Ren, "Using GAN Neural Networks for Super-Resolution Reconstruction of Temperature Fields," *Intelligent Automation & Soft Computing*, vol. 35, no. 1, pp. 941-956, Jun. 2023. [Article \(CrossRefLink\)](#)
- [27] G. Accarino, M. Chiarelli, F. Immorlano, V. Aloisi, A. Gatto and G. Aloisio, "Msg-gan-sd: A multi-scale gradients gan for statistical downscaling of 2-meter temperature over the euro-cordex domain," *AI*, vol. 2, no. 4, pp. 600-620, Nov. 2021. [Article \(CrossRefLink\)](#)
- [28] T. Izumi, M. Amagasaki, K. Ishida and M. Kiyama, "Super-resolution of sea surface temperature with convolutional neural network-and generative adversarial network-based methods," *J. Water Clim. Chang.*, vol. 13, no. 4, pp. 1673-1683, Mar. 2022. [Article \(CrossRefLink\)](#)
- [29] J. KimT. Kim and J. Ryu, "Multi-source deep data fusion and super-resolution for downscaling sea surface temperature guided by Generative Adversarial Network-based spatiotemporal dependency learning," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 119, pp. 103312, May. 2023. [Article \(CrossRefLink\)](#)
- [30] O. Miralles, D. Steinfeld, O. Martius and A.C. Davison, "Downscaling of Historical Wind Fields over Switzerland Using Generative Adversarial Networks," *Artificial Intelligence for the Earth Systems*, vol. 1, no. 4, pp. e220018, Nov. 2022. [Article \(CrossRefLink\)](#)
- [31] Y. Yang, Q. You, Z. Jin, Z. Zuo and Y. Zhang, "Construction of surface air temperature over the Tibetan Plateau based on generative adversarial networks," *Int. J. Climatol.*, vol. 42, no. 16, pp. 10107-10125, Oct. 2022. [Article \(CrossRefLink\)](#)
- [32] C. Ge, Z. Liu, W. Susilo, L. Fang and H. Wang, "Attribute-Based Encryption with Reliable Outsourced Decryption in Cloud Computing Using Smart Contract," *IEEE Transactions on Dependable Secur Computing*, vol. 21, no. 2, pp. 937-948, 2024. [Article\(CrossRefLink\)](#)
- [33] C. Ge, W. Susilo, Z. Liu, J. Baek, X. Luo and L. Fang, "Attribute-Based Proxy Re-Encryption With Direct Revocation Mechanism for Data Sharing in Clouds," *IEEE Transactions on Dependable Secur Computing*, vol. 21, no. pp. 949-960, 2024. [Article\(CrossRefLink\)](#)
- [34] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia and L. Fang, "Revocable attribute-based encryption with data integrity in clouds," *IEEE Transactions on Dependable Secur Computing*, vol. 19, no. 5, pp. 2864-2872, Sept.-Oct. 2022. [Article\(CrossRefLink\)](#)
- [35] J. HuL. Shen and G. Sun, "Squeeze-and-excitation networks," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 7132-7141, 2018. [Article\(CrossRefLink\)](#)
- [36] M. Siam, S. Valipour, M. Jagersand and N. Ray, "Convolutional gated recurrent networks for video segmentation," in *Proc. of 2017 IEEE international conference on image processing (ICIP)*, pp. 3090-3094, 2017. [Article\(CrossRefLink\)](#)
- [37] W. Shi, et al, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 1874-1883, 2016. [Article\(CrossRefLink\)](#)
- [38] M. ArjovskyS. Chintala and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. of International conference on machine learning(ICML)*, pp. 214-223, 2017. [Article\(CrossRefLink\)](#)
- [39] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings*, pp. 249-256, 2010. [Article\(CrossRefLink\)](#)
- [40] Y. BengioP. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157-166, Mar. 1994. [Article \(CrossRefLink\)](#)
- [41] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen," *Diploma, Technische Universität München*, vol. 91, no. 1, 1991. [Article \(CrossRef Link\)](#)
- [42] H. Zhao, O. Gallo, I. Frosio and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imaging*, vol. 3, no. 1, pp. 47-57, Mar. 2017. [Article\(CrossRefLink\)](#)

- [43] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600-612, Apr. 2004. [Article \(CrossRefLink\)](#)
- [44] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *arXiv:1706.08500*, Dec. 2017. [Article \(CrossRefLink\)](#)



Tao Li: He is an associate Professor at School of Artificial Intelligence, Nanjing University of Information Science & Technology. He received the B.S. and M.S. degree in Nanjing University of Information Science & Technology, and PHD in Nanjing University of Aeronautics and Astronautics. His research interests mainly include machine learning, big data, data mining.



Liang Wang: He is currently pursuing a master degree at the school of computing in Nanjing University of Information Science & Technology, Nanjing, China. His current research interests include data mining and machine learning.



Li Na Wang: She is an associate Professor at School of artificial intelligence, Nanjing University of Information Science & Technology. she received the B.S. and M.S. degree in Nanjing University of Information Science & Technology, and PHD in Nanjing University of Aeronautics and Astronautics. Her research interests mainly include data mining, artificial intelligence and network security.