Using Conditional Generative Adversarial 3-D Convolutional Neural Network for Precise Radar Extrapolation

Cong Wang, Ping Wang, Pingping Wang, Bing Xue, and Di Wang

***Abstract*—Radar echo extrapolation is a basic but essential task in meteorological services. It could provide radar echo prediction results with high spatiotemporal resolution in a computationally efficient way, and effectively enhance the operational system's forecasting capability for meteorological hazards. Traditional methods perform extrapolation by estimating echo motions between contiguous radar data. This strategy is difficult to characterize complex nonlinear meteorological processes effectively, and it is difficult to benefit from large historical data. Recently, machine learning (ML) models have been used for radar echo extrapolation. These methods have effectively improved extrapolation quality in a data-driven way and from the statistical perspective. Although the ML-based methods show excellent performance, they usually produce blurry extrapolations. This leads to underestimating radar echo intensity and making echo lack small-scale details. Moreover, it makes models difficult to predict severe convective hazards. To solve this problem, a two-stage extrapolation model based on 3-D convolutional neural network and conditional generative adversarial network is proposed. These two models form the "pre-extrapolation" and "postprocessing" paradigm. The pre-extrapolation model is trained in the traditional way and performs rough extrapolation. The postprocessing model uses the pre-extrapolation result as input and is trained with the adversarial strategy. It could correct the echo intensity and increase the echo's details. In the experiment, our model could provide more precise radar echo extrapolations than other methods, especially for intense echoes and convective systems, in the data of North China from 2015 to 2016.**

***Index Terms*—Conditional generative adversarial network (CGAN), convective system, convolutional neural network (CNN), radar extrapolation.**

I. Introduction

WEATHER radar is one of the vital weather observation tools. It can provide high-precision atmospheric information in real-time and is also the basis of many weather recognition and forecasting algorithms [1]–[5]. Radar echo extrapolation technology refers to predicting future radar echo changes based on radar data in the past time. This technology analyzes radar echo changes in the past few moments and can provide high spatiotemporal resolution future atmospheric evolution information within a few minutes after receiving the data. Extrapolation technology has solid forecast skill in the next few hours, especially the first hour. High quality radar echo extrapolation result is the cornerstone of many forecasting algorithms [6]–[10] and is of great significance to disaster prevention and control.

Traditional radar extrapolation methods could be divided into cell-based methods [11]–[15] and pixel-level methods [16]–[20]. The thunderstorm identification, tracking, analysis, and nowcasting (TITAN) method [11] and the storm cell identification and tracking (SCIT) method [12] are representatives of cell-based methods. These methods take the storm cells as objects and calculate each cell's motion vector by tracking its centroid. The predicted trajectory of each cell is usually calculated by fitting or weighted summation of its historical motion vectors. However, pixel-level methods attempt to obtain dense motion vector fields and perform extrapolation point-by-point. Tracking the radar echo with correlation (TREC) method [16] is representative of pixel-level methods. It calculates the motion vector based on the correlation between the neighborhood of each point. The optical flow method in image processing is also used for radar extrapolation [21], [22]. This method can be regarded as a particular case of cross-correlation methods, which also obtains a dense motion vector field through pixel-level matching. The advantage of the optical flow method is that its principle has already included the global smoothing constraint, which could reduce the noise in motion vector fields more effectively. After determining the dense motion vector field, the extrapolation of each point is usually determined by the forward or backward semi-Lagrangian scheme [21], [23]. Different from the above perspectives, some studies have tried to handle extrapolation from the signal processing view. These studies analyzed the dependence between radar echo scale and predictability and filtered small scale low predictability echoes to improve the forecast skill [24]–[27]. Moreover, additional random noise was used to reconstruct the uncertainty caused by the small-scale structure [28].

Although all of these methods have provided radar extrapolation solutions, some weaknesses still need to be addressed. Cell-based methods could retain the cells' structure during extrapolation, but they could not cover the radar echo without any cells. Pixel-level methods could perform pointwise extrapolation, but

Manuscript received January 9, 2021; revised March 22, 2021 and April 25, 2021; accepted May 18, 2021. Date of publication May 25, 2021; date of current version June 11, 2021. This work was supported by the China Special Fund for Meteorological Research in the public interest under Grant GYHY201406004. *(Corresponding author: Di Wang.)*

Cong Wang, Ping Wang, Pingping Wang, and Di Wang are with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China, and also with Joint Laboratory of Intelligent Identification and Nowcasting Service for Convective System, CMA Public Meteorological Service Center, Beijing 100081, China (e-mail: wangc@tju.edu.cn; wangps@tju.edu.cn; wpp1992@tju.edu.cn; wangdi2015@tju.edu.cn).

Bing Xue is with the CMA Public Meteorological Service Center, Beijing 100081, China (e-mail: xuebing2009@163.com).

Digital Object Identifier 10.1109/JSTARS.2021.3083647

the neighborhood scale and smoothing constraint make it more challenging to predict complex local evolution, especially for convective systems. The signal processing-based methods are also incapable of handling local precision extrapolation while filtering out low predictability radar echoes. Moreover, these traditional methods usually treat the echo intensity as a constant during extrapolation, but the actual echo is always accompanied by enhancement and weakness. Furthermore, they only use the current radar sequence for extrapolation and could not benefit from a mass of historical samples. All these flaws determine that the traditional extrapolation method is difficult to forecast the complex nonlinear meteorological process accurately.

In recent years, machine learning (ML) methods have been widely used in various fields of meteorology and have achieved remarkable results [2], [29]–[34]. ML methods could effectively represent complex nonlinear meteorological processes from the statistical view by modeling large historical data. In this way, a series of ML-based radar echo forecasting methods have been proposed [35]. Shi *et al.* [36] first proposed an end-to-end radar extrapolation method (ConvLSTM) for precipitation nowcasting. This method used a convolutional neural network (CNN) to extract features in radar images and used a long short term memory (LSTM) network to model the time dependence between radar scans. Compared with the optical flow method, it has achieved significant performance improvement. Furthermore, Shi *et al.* [37] improved this method and proposed the trajectory gated recurrent unit (TrajGRU) model, which enhances the model's spatiotemporal representation capability. Contrarily, Han *et al.* [38] paid more attention to the convective systems. He divided radar mosaic images into boxes sized $6\,\mathrm{km} \times 6\,\mathrm{km}$. For each box, a CNN model was used to extract features from the 3-D grid radar data and meteorological reanalysis data. This method finally aimed to predict whether each box's reflectivity would be greater than 35 dBZ in the next 30 minutes. Tran and Song [39] combined 3-D CNN (3D-CNN) and gated recurrent unit (GRU) to extrapolate 3-D radar data and designed a new module to solve the data crosstalk problem between different channels.

ML-based radar extrapolation methods have made significant progress compared with traditional algorithms. However, there are still some challenges. When constructing an ML model, a loss function that could measure the difference between the prediction and the ground truth is designed, and the ML model performance is improved by minimizing this loss function. Almost all ML-based radar extrapolation methods employed mean square error (MSE) and its extensions as the loss function. However, some studies that used ML for image spatiotemporal prediction have shown that MSE usually "blur" the results [40]–[44]. The reason for this "blur" could be explained from two perspectives. 1) MSE is sensitive to outliers, which makes the model more inclined to mean prediction [43], [44]. 2) The predictability of radar echo is related to the echo scale, and small-scale high-frequency echo details usually have low predictability [24]–[27], [45]. For the radar extrapolation task, this blur phenomenon will produce two adverse effects on extrapolations. 1) The reflectivity intensity of the extrapolations are usually underestimated. 2) The echo small-scale details in the extrapolations are lost, and the convective systems' structure cannot be represented correctly. These make it difficult for ML methods to provide sufficient information for hazard forecasting, especially for convective hazards.

In the computer vision field, generative adversarial network (GAN) is usually regarded as a method capable of reducing the blur phenomenon brought by MSE and enhancing image detail and sharpness. The GAN is a kind of model that was originally used to generate images from given data [46]. It consists of two parts. The first part is called the generator, which is responsible for generating the required images. The other part is called the discriminator, which is responsible for estimating the authenticity of the generator outputs. The discriminator could be regarded as a loss function with learning ability and is updated with the generator's training, which could dynamically measure the quality of images generated by the generator. GAN could overcome the problems caused by specific loss functions and can generate realistic images. Some recent researches have begun to investigate combining GAN to reduce the blur phenomenon in ML-based extrapolation [47]–[49]. They applied the naive GAN paradigm directly to the extrapolation models' training process and improved the pixel-level score. However, limited by the naive GAN's inherent characteristics, GAN models are difficult to generate high-resolution images. Therefore, it is challenging for these studies to predict echoes with accurate structures and small-scale details precisely. [49].

In this article, a two-stage model combining 3-D-CNN and conditional generative adversarial network (CGAN) is proposed to tackle these problems. Compared with GAN, CGAN extra joins explicit constraints between the input and output of the generator [50]. However, in this article, CGAN has not been used to undertake all the radar extrapolation jobs. Radar data is high-dimensional spatiotemporal data, and training GAN-based models with high-dimensional data is still a challenge [51]–[53]. Because the space of the generated results expands drastically with the increase of dimension, it is difficult for the GAN-based models to converge. From the meteorological view, it is difficult for the models to precisely predict locations and intensities of radar echoes simultaneously with a high resolution during adversarial training. Some researches have made improvements through joint training of multiple discriminators of different scales [54], [55] or step-by-step training at multiple resolutions [56]–[59]. In this article, a 3-D-CNN-based "pre-extrapolation" model using an enhanced MSE as the loss function is first trained. Subsequently, the pre-extrapolation model's output is used as input to train a CGAN-based "postprocessing model." From the ML perspective, the pre-extrapolation model can be regarded as the pretraining model of the postprocessing model and making the CGAN-based postprocessing easier to converge. From the operational application perspective, the radar extrapolation problem is decoupled into two parts. The pre-extrapolation model is responsible for exactly predicting echo location and roughly extrapolating the echo intensity and distribution. The postprocessing model is responsible for fine-tuning the intensity and details to produce the final realistic extrapolation result.

This article is organized as follows. Section II shows the data used in this article. Section III describes the method proposed in

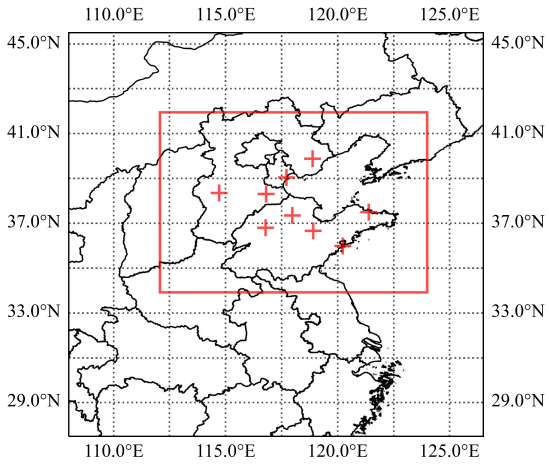

Fig. 1. Study area and radar locations.

this article. The results and case studies for different applications are shown in Section IV. Section V is the discussion. The summary and future work are in section VI.

II. Data

The *S*-band doppler weather radar data provided by the Meteorological Observation Center of the China Meteorological Administration are used for radar echo extrapolation. These radars are located in Qinhuangdao, Tianjin, Shijiazhuang, Cangzhou, Binzhou, Jinan, Weifang, Qingdao, and Yantai in North China. A general view of the study area and radar locations is shown in Fig. 1. The time range is from May 2015 to December 2016. The radar scan is performed approximately every 6 minutes, and the scan range is $230\,\mathrm{km}$. Each scan includes nine elevation angles from 0.5° to 19.5°. The radar data are converted from polar coordinates to cartesian coordinates with a resolution of $2\,\mathrm{km} \times 2\,\mathrm{km}$. The composite reflectivity image product of weather radar is used for echo extrapolation. Every 20 consecutive radar scans are considered as a complete sample. The first 10 radar images are used as the model's input to predict the next 10 radar images about one hour in the future. The last 10 radar images are used as labels to evaluate model performance. There is no time overlap between samples. A total of 17 672 samples are collected. The samples are divided into a training set, a validation set, and a test set in terms of date to ensure the independence between data sets. The samples earlier than the 20th of each month are used as the training set to train the radar extrapolation model. The radar data from the 21st to the 25th of each month are regarded as the validation set for model tuning. After the 26th of each month, the samples form a test set to evaluate the final performance. The training set, validation set, and test set contain 10 900, 3497, and 3275 samples, respectively.

III. Method

In this article, the proposed method consists of two parts. 1) The 3-D-CNN-based radar echo pre-extrapolation model. 2) The CGAN-based radar echo postprocessing model. The first part is handled for the extrapolation of echo position and rough prediction of the echo intensity and distribution in the future. The second part is responsible for postprocessing the first part results so that the final results are closer to the authentic radar echo in shape detail, and intensity. The method flow chart is shown in Fig. 2.

A. Three-Dimensional-CNN-Based Pre-Extrapolation Model

1) Three-Dimensional Convolution Kernel: In recent years, CNN has made significant progress in image processing. Thanks to CNN's structure conforming to the characteristics of image data. Image data usually denoted as a 2-D matrix with size $H \times W$ (H denotes the image height, W denotes the image width). Correspondingly, the traditional 2-D convolution kernels also perform feature extraction in this 2-D space. Compared with naive image processing problems, radar extrapolation can be seen as adding the time correlation to spatial dimensions. For spatiotemporal data, it is usually represented as a 3-D grid of $T \times H \times W$ (T denotes the time dimension). If using 2-D convolution to process spatiotemporal data directly, the feature extraction could only be performed in a 2-D space of $H \times W$. And the time dimension is regarded as the number of image channels and will not perform any additional processing [see Fig. 3(a)]. To enhance the model's ability for extract features in the time dimension, the 3-D convolution kernel could be obtained by expanding the 2-D convolution kernel in the time dimension [60], [61]. The 3-D kernel regards time dimension and space dimension equally important and performs feature extraction in the 3-D space of $T \times H \times W$ [see Fig. 3(b)]. In this article, the 3-D convolution kernel is used as the basic unit to extract spatiotemporal features from radar data.

2) Model Structure and Details: The model consisting of 3-D convolution kernels is used for the preliminary extrapolation. The model structure is shown in Fig. 4. This model consists of three kinds of blocks: encoding, inference, and decoding.

The encoding blocks are responsible for encoding the spatiotemporal radar data from high spatiotemporal resolution into a low-dimensional space. In this process, it not only obtains more powerful features but also reduces computing resources. The inference blocks are responsible for further extracting advanced features from the encoded low-dimensional features and performing spatiotemporal prediction. In this process, the dimensions of the features remain consistent. The decoding blocks are used for reconstructing the low-dimensional features to the original radar space. There are four encoding blocks, each of which consists of two 3-D convolution kernels and an additional 3-D downsampling layer. Each time the data passes through an encoding block, the spatial resolution becomes halved, and the number of channels is doubled. After the last encoding block, the time resolution is also downsampled. Subsequently, there are five inference modules, each of which consists of a 3-D convolution kernel that does not follow any sampling layer. The number of decoding modules is the same as that of encoding modules. Each decoding module consists of two 3-D convolution kernels and an additional 3-D upsampling layer. Each time the data passes through a decoding module, the spatial resolution

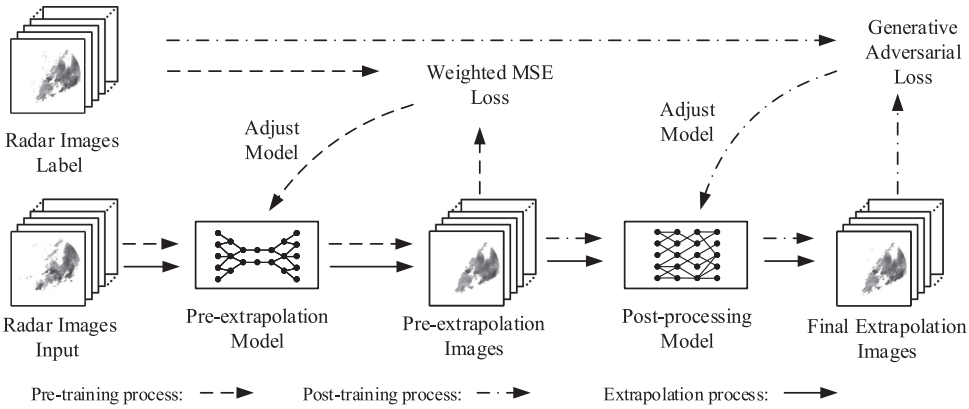

Fig. 2. Method flow chart. The dotted line represents the training process of the pre-extrapolation model. The dashdotted line indicates the training process of the postprocessing model. The solid line represents the working sequence of the models in operational applications. During the pre-extrapolation model training, past radar images are used as model input, and the weighted MSE loss function is used to update the model parameters. The pre-extrapolation model result is used as the input of the postprocessing model, and the generative adversarial loss is used to train the postprocessing model. In operational applications, the pre-extrapolation model and the postprocessing model are executed sequentially to produce the final radar extrapolation result.

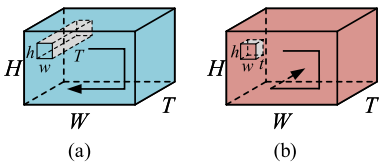

Fig. 3. $h \times w$ 2-D convolution kernel and a $t \times h \times w$ 3-D kernel are used to process spatiotemporal data of size $T \times H \times W$. (a) 2-D convolution traverses 3-D data along the $H \times W$ 2-D space. (b) 3-D convolution traverses data in both time and space dimensions.

is restored by $\times 2$. The first decoding module is additionally responsible for the restoration of time resolution. The last decoding module consists of an upsampling layer and a 3-D convolution kernel. At the end of the model, a naive 3-D convolution kernel is used to transform the decoding model output into the image space. The specific model parameters are shown in appendix Table A1.

Additionally, some valuable techniques are also used to build the model. The batch normalization (BN) [62] is used after each convolution kernel. It enables the data to be adaptively normalized after each convolution kernel and retain the original distribution, which can effectively accelerate model convergence. The leaky rectified linear unit (LeakyRELU) [63] with a slope of 0.2 is also used after each convolution layer to enhance the model's nonlinear modeling ability. The last layer is only used to transform the number of channels, and the BN and the LeakyRELU are not used.

3) Loss Function: Because it only needs to estimate the intensity and distribution of radar echoes roughly, MSE is used to train the pre-extrapolation model. In radar data, higher reflectivity usually means more severe weather hazards. However, data with high reflectivity accounts for a small proportion of all radar data [37]. This means that the model will focus on improving the accuracy of prediction in weak reflectivity areas. To alleviate this problem, a weighted MSE (WMSE) loss function is used to optimize the pre-extrapolation model (1)–(2). Higher reflectivity is given a higher weight, which allows the model to pay more attention to areas prone to severe meteorological hazards

$$L_{\mathrm{WMSE}}\left(I, \hat{I}\right) = \sum_{h,w,t} weight_{h,w,t} \times \left(I_{h,w,t} - \hat{I}_{h,w,t}\right)^2 \tag{1}$$

$$weight_{h,w,t} = \begin{cases} 1, & if \quad I_{h,w,t} < 30\,\mathrm{dBZ} \\ 2, & if \quad 30\,\mathrm{dBZ} \le I_{h,w,t} < 35\,\mathrm{dBZ} \\ 5, & if \quad 35\,\mathrm{dBZ} \le I_{h,w,t} < 40\,\mathrm{dBZ} \\ 10, & if \quad 40\,\mathrm{dBZ} \le I_{h,w,t} < 45\,\mathrm{dBZ} \\ 30, & if \quad 45\,\mathrm{dBZ} \le I_{h,w,t} \end{cases} \tag{2}$$

where $I_{h,w,t}$ represents the reflectivity of the label data with coordinate (h, w, t), $\hat{I}_{h,w,t}$ represents the prediction data reflectivity with coordinate (h, w, t), and $weight_{h,w,t}$ represents the weight of with coordinate (h, w, t).

B. CGAN-Based Post-Processing Model

The pre-extrapolation model could give a rough prediction of radar intensity distribution at accurate location but lacks local shape details and still has deviations in intensity. This hinders accurate forecasts, especially for convective systems, which usually cause severe hazards. In this section, a CGAN-based postprocessing model is designed to improve the accuracy of the pre-extrapolation model results.

1) CGAN Method: GAN is an algorithm proposed by Goodfellow *et al.* [46] to generate realistic images from given data. It reduces the "blur" effect caused by the pixel-based loss function in traditional algorithms. GAN transforms the traditional "model-loss" paradigm into the "generator-discriminator" paradigm. The generator uses the given data to generate spurious images. However, it does not compare the generated images and the label images pixel by pixel to calculate the loss. Instead, the generated images and the label images are input into a new model

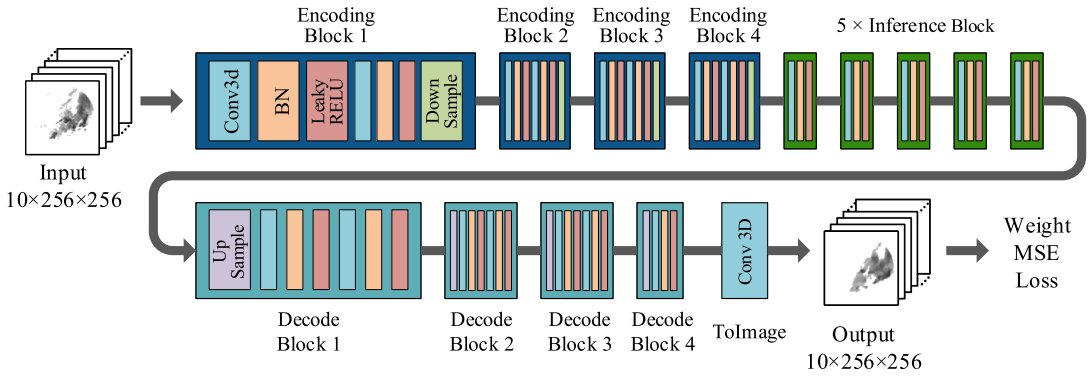

Fig. 4. Structure of the 3-D-CNN-based pre-extrapolation model.

called "discriminator," and the discriminator is used to determine inputs' authenticity. The training process of GAN can be seen as an adversarial game between the generator and the discriminator. The generator's purpose is to make the discriminator unable to distinguish the authenticity of the images generated by the generator and the label images. The discriminator is designed to identify the generator's outputs from label images. These two models are trained by the established goals, iteratively. Finally, a generator capable of generating realistic images and a discriminator competent in judging images' authenticity can be obtained. GAN breaks the limitation introduced by the loss function in traditional image generation models. It uses the discriminator to supervise and improve the generation model to replace the specific loss function. Because CNN could effectively extract high-level features of images, the discriminator is usually composed of CNNs. For the discriminator, what needs to be processed is only an image classification problem. The training process of GAN can be defined as an alternate maximum and minimum optimization process of the classification loss. The loss is defined as

$$L_{\mathrm{GAN}}(G,D) = E_{\boldsymbol{x}\sim set_{data}}\left[logD(\boldsymbol{x})\right] + E_{\boldsymbol{z}\sim set_{\boldsymbol{z}}}\left[logD(1-D\left(G\left(\boldsymbol{z}\right)\right))\right] \tag{3}$$

where G and D represent the generator and the discriminator, respectively. E represents the mathematical expectation. $\boldsymbol{x} \sim set_{data}$ means that $\boldsymbol{x}$ is sampled in the set_{data}. $\boldsymbol{x}$ denotes a real image, set_{data} is the set of label images, $\boldsymbol{z}$ represents given information used to generate images, and $set_{\boldsymbol{z}}$ is the set of $\boldsymbol{z}$. Usually, the generator and the discriminator are composed of neural networks, and the batch stochastic gradient descent algorithm is used to update models. The iterative optimization steps are shown in Algorithm 1.

The original GAN is used to generate realistic images from noise. There are no explicit constraints between the input and output of the generator. However, in applications, the generator is always expected to complete tasks based on specific input (e.g., the input is a low-resolution image, and the output is the corresponding high-resolution image) is always more desired. To solve this problem, CGAN is proposed [50]. Unlike the GAN, the CGAN's discriminator inspects whether the generated image matches the generator input instead of simply judging

Algorithm 1: Iteratively Optimize GAN Using Batch Stochastic Gradient Descent Method.

Definition:G represents the generator with parameters $\boldsymbol{\theta}_g$; D denotes the discriminator with parameters $\boldsymbol{\theta}_d$.

1: **for** each iteration **do**

2: Sample batch of m inputs $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m\}$ from $set_{\boldsymbol{z}}$ for generating images.

3: Sample batch of m actual images $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$ from set_{data} for labels.

4: Update the parameterd $\boldsymbol{\theta}_d$ of the discriminator D to maximize $L_{\mathrm{GAN}}(G,D)$:

$$\boldsymbol{\theta}_d := \boldsymbol{\theta}_d + \nabla\boldsymbol{\theta}_d \frac{1}{m}\sum_{i=1}^{m}[logD(\boldsymbol{x}_i) + log(1-D(G(\boldsymbol{z}_i)))].$$

5: Update the parameterd $\boldsymbol{\theta}_g$ of the generator G to minimize $L_{\mathrm{GAN}}(G,D)$:

$$\boldsymbol{\theta}_g := \boldsymbol{\theta}_g - \nabla\boldsymbol{\theta}_g \frac{1}{m}\sum_{i=1}^{m} log(1-D(G(\boldsymbol{z}_i))).$$

6: **end for**

the generator output's authenticity. Therefore, the input and output of the generator are used together as the input of the discriminator. The loss function of CGAN is as follows:

$$L_{\mathrm{CGAN}}(G,D) = E_{\boldsymbol{c}\sim set_{\boldsymbol{c}},\boldsymbol{x}\sim set_{data}}\left[logD(\boldsymbol{c},\boldsymbol{x})\right] + E_{\boldsymbol{c}\sim set_{\boldsymbol{c}}}\left[logD(1-D\left(\boldsymbol{c},G\left(\boldsymbol{c}\right)\right))\right] \tag{4}$$

where $\boldsymbol{c}$ represents the condition for generating images, $set_{\boldsymbol{c}}$ is the set of $\boldsymbol{c}$, and other symbols are the same as those in (3). The optimization process has also changed (see appendix Figure A1).

In this article, CGAN is used to enhance the pre-extrapolation model results. The generator takes the pre-extrapolation model results as input and outputs the corresponding radar images with more details. The discriminator executes to determine whether the output and input of the generator are matched. Pre-extrapolation results and real radar images are joined used as positive samples of the discriminator, and the expected output of the discriminator is "True" or "1." The negative samples of the discriminator are the pre-extrapolated radar images and the

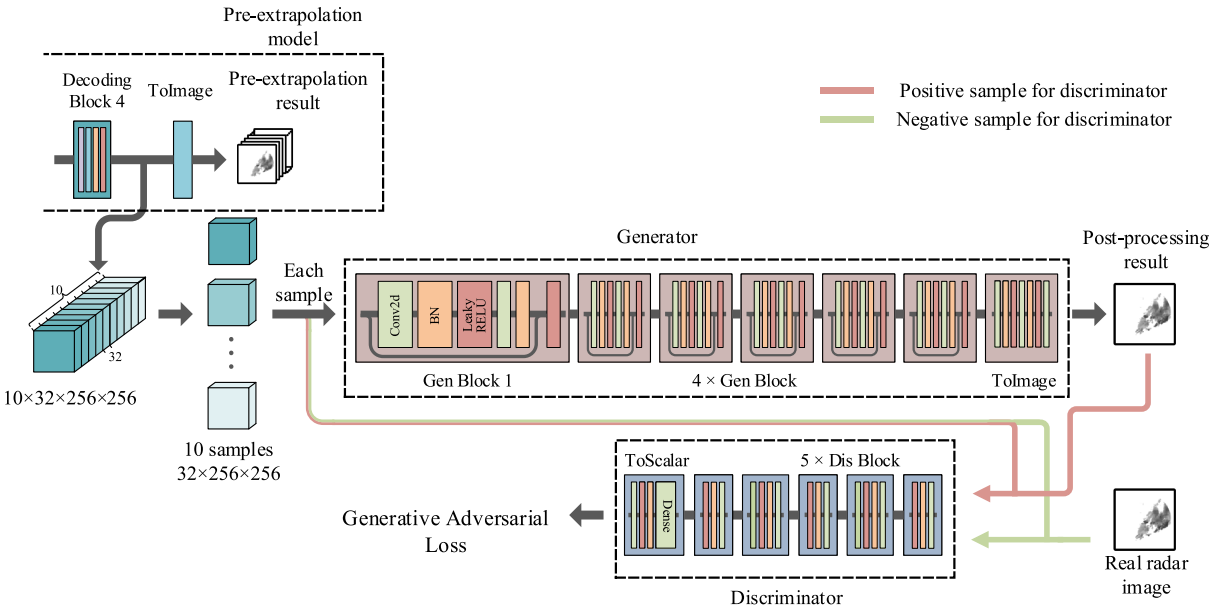

Fig. 5. Structure of the CGAN-based postprocessing model.

generated radar images, and the expected output is "False" or "0." The generator and discriminator are optimized iteratively. Ultimately, a postprocessing model capable of outputting accurate radar images based on the pre-extrapolation results can be obtained.

2) Model Structure and Details: The postprocessing model structure is shown in Fig. 5. The pre-extrapolation model results are sent to the generator one by one to generate more accurate radar images. However, the generator inputs are not the radar images generated by the pre-extrapolation model but the feature maps of the last decode block in the pre-extrapolation model. This is because the outputs after the last decoding block contain more latent information than the radar images. In this way, each sample of the generator is a 256×256 image with 32 channels. The generator consists of 14 layers of the 2-D kernel. Each 2 of the first 12 layers together with BN and LeakyRELU form a generator block, which does not include any sampling operations. The residual concept is also introduced to enhance generator performance [64]. The generator's input and output are connected as the negative sample of the discriminator, and the generator's input and the label images are paired as the positive sample of the discriminator. The resolution of both positive and negative samples is $33 \times 256 \times 256$. The discriminator is composed of five layers of 2-D convolution layers and two fully connected layers. Each 2-D convolution is accompanied by BN and LeakyRELU to form a discriminator block. After every two discriminator blocks, downsampling is performed. The final fully connected layer gives a scalar output, which is used as the score of the discriminator input. The parameters of the generator and the discriminator are shown in appendix Table A2.

3) Loss Function: Equation (4) shows the original loss function of CGAN. However, it is challenging to train CGANs using the basic loss function, and the improvement of the loss function is also one of the hotspots of ML research. In this article, for training the CGAN model, the loss function based on Wasserstein distance [65], [66] is employed (5). The discriminator based on this loss function no longer estimates the positive and negative of samples (output between 0 and 1) but judges the degree to which the input belongs to a positive sample or a negative sample (output range is infinity). Additionally, the Lipschitz continuity is also required by the discriminator, and a penalty term based on the Lipschitz continuity is added to the loss function. The $set_{\hat{\boldsymbol{c}},\hat{\boldsymbol{x}}}$ represents a set, which is composed of points between labels and generated images

$$\begin{aligned} L_{\mathrm{CGAN}}(G,D) = & E_{\boldsymbol{c}\sim set_{\boldsymbol{c}},\boldsymbol{x}\sim set_{data}}\left(D(\boldsymbol{c},\boldsymbol{x})\right) \\ & - E_{\boldsymbol{c}\sim set_{\boldsymbol{c}}}\left(D(c,G(\boldsymbol{c}))\right) \\ & - \lambda E_{\hat{\boldsymbol{c}},\hat{\boldsymbol{x}}\sim set_{\hat{\boldsymbol{c}},\hat{\boldsymbol{x}}}}\left(\left\|\nabla_{\hat{\boldsymbol{c}},\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{c}},\hat{\boldsymbol{x}})\right\|_2 - 1\right)^2 \end{aligned} \tag{5}$$

IV. Experiments and Results

In this section, the model performance is evaluated. Pixel-based and convective cell-based evaluations are used to illustrate model performance in different meteorological applications. The polynomial-based Farneback optical flow method [67], TrajGRU [37], and simple 3-D-CNN are used as baselines. Our extrapolation GAN model is denoted as ExtGAN. Farneback optical flow uses local polynomial matching for dense motion vector field estimation, and the forward-in-time scheme proposed in [21] is used as the corresponding extrapolation method. TrajGRU is an effective method that combines gated recurrent unit (GRU) and 2-D-CNN for radar extrapolation. The simple 3-D-CNN is used to illustrate the role of the CGAN model. Additionally, to illustrate the role of WMSE, a 3-D-CNN w/o WMSE model is trained and compared with the 3-D-CNN. All models are trained with the training set, the validation set is

TABLE I
CONTINGENCY TABLE OF INDICATORS FOR MODEL EVALUATION

	Observed yes	**Observed no**
Predicted yes	a	b
Predicted no	c	d

applied for model hyperparameters adjustment, and the test set is used to evaluate models' final performance.

1) Evaluation Indicators: In experiments, the method based on the contingency table is used to evaluate models at pixel-level. This method sets different reflectivity thresholds, which could indicate model performance under different levels of hazards. When a threshold is set, every point on the predicted radar image is transformed into "predicted yes" or "predicted no," and every point on the label image is transformed into "observed yes" or "observed no." As shown in Table I, a, b, c, and d represent the number of different type results. Based on these, five indicators are calculated (6)–(10). Probability of detection (POD) indicates how many hazards are correctly predicted. False alarm ratio (FAR) indicates the ratio of the predicted hazards that do not occur. Equitable threat score (ETS) gives a more comprehensive evaluation of hits through penalizing misses and false alarms and adjusting hits associated with random chance. Bias indicates whether the model has over-prediction or under-prediction. Heidke skill score (HSS) represents the advancement of model results relative to random forecast

$$POD = \frac{a}{a+c} \tag{6}$$

$$FAR = \frac{b}{a+b} \tag{7}$$

$$Bias = \frac{a+b}{a+c} \tag{8}$$

$$HSS = \frac{2\left(ad-bc\right)}{\left(a+c\right)\left(c+d\right)+\left(a+b\right)\left(b+d\right)} \tag{9}$$

$$ETS = \frac{a-a_{ref}}{a-a_{ref}+b+c}, a_{ref} = \frac{(a+b)(a+c)}{a+b+c+d}. \tag{10}$$

Convective systems usually cause severe weather hazards. Convective cells are the basic units of convective systems and also the basic objects for convective hazard research and forecast. To exhibit the prediction ability of extrapolation methods for convective systems, two statistical indicators based on convective cell intensity are used to evaluate model performance. The maximum reflectivity (MR) and the average reflectivity (AR) of each convective cell in label radar images and extrapolation results are counted separately. The former could show the forecast skill of models for the most severe hazard in each convective cell. The latter could show the accuracy of extrapolation models in estimating the average intensity of convective cells. Furthermore, the number of cells included in extrapolations are also counted to evaluate model performance for convective hazard forecasting. In experiments, in order to prevent the interference caused by different cell identification methods, the convective cell is defined as a region larger than $20\,\mathrm{km}^2$ with reflectivity greater than 40 dBZ.

2) Pixel-Level Results: The pixel-level evaluation method is first performed. Table II shows the ETS, bias, and HSS scores of the models averaged over all moments on the test set with different reflectivity thresholds. For the baselines and ExtGAN, ETS and HSS illustrate the performance of the model directly. The optical flow model always gets the lowest score at low reflectivity thresholds but has better performance at 50 dBZ. On the contrary, the ML models have better performance at the low reflectivity threshold but have different achievements at 50 dBZ. At lower reflectivity thresholds, ExtGAN's performance is similar to that of the others. As the reflectivity threshold increases, these models' performance deteriorates, but ExtGAN declines the slowest. When the threshold reaches 50 dBZ, our model shows a significant advantage. 50 dBZ usually means severe hazards. ExtGAN still has acceptable forecast skill (ETS is 0.1002, HSS is 0.1821), but other ML models get poor performance (ETS and HSS are both in 10^{-2} magnitude). Bias score shows models' ability for hazard scale forecasting. When the reflectivity threshold is between 20 dBZ and 40 dBZ, the bias score of the optical flow and the 3-D-CNN is less than 1, which tends to under-forecasting the hazard; the bias score of the TrajGRU is greater than 1, which tends to over-forecasting; ExtGAN's score is around 1, which means that the scale of the hazard forecast is close to the real scale. When the reflectivity threshold is 50 dBZ, ExtGAN's bias score is 0.7616, and the scores of other ML models are around 10^{-2} magnitude. The trends of the bias scores of these models are also different. The bias score of the optical flow and 3-D-CNN w/o WMSE decreases as the threshold increases. However, the other ML models' bias scores all increase first and then decrease. Additionally, the 3-D-CNN w/o WMSE just has a performance similar to 3-D-CNN with the 20 dBZ threshold (see Table II). At 30 and 40 dBZ, the performance of 3-D-CNN w/o WMSE is far worse than that of 3-D-CNN. This shows that WMSE could improve the model's forecast skill for intense echoes.

More complete pixel-level evaluation results are shown in Fig. 6. The magnitude relationship between all scores is consistent with that in Table II. ETS, POD, and HSS have similar trends, and they all decrease with time. The optical flow model always has an excellent performance in the first few prediction times, but as time increases, the performance drops sharply. The TrajGRU model shows a similar trend to the optical flow. The 3-D-CNN and ExtGAN are more stable. For the FAR score, the models all increase with time, but our model variations are more stable. The bias scores of both optical flow and ExtGAN decrease with time increase. On the contrary, the bias scores of the other models gradually increase with time.

3) Convective Cell-Based Results: Two convective cell-based evaluation indicators are performed. Table III shows the MR and AR statistics of different models on all test set samples. For the MR statistics, the label radar images are higher than all extrapolation models. ExtGAN reports the highest MR among all comparison models. Also, ExtGAN's statistic is the closest to label radar images. In baseline models, 3-D-CNN is the closest to the label radar image. The MRs of all models at different times

TABLE II
ETS, BIAS, AND HSS SCORES OF THE MODELS AVERAGED OVER ALL MOMENTS WITH THE THRESHOLDS OF 20, 30, 40, AND 50 DBZ

Algorithms	ETS				Bias				HSS			
	20dBZ	30dBZ	40dBZ	50dBZ	20dBZ	30dBZ	40dBZ	50dBZ	20dBZ	30dBZ	40dBZ	50dBZ
Optical flow	0.4596	0.3311	0.1706	0.0725	0.6593	0.5574	0.4030	0.2035	0.6298	0.4975	0.2915	0.1353
3D-CNN	0.5590	0.4387	0.2337	0.0221	0.7864	0.8055	0.5224	0.0511	0.7172	0.6098	0.3788	0.0433
3D w/o WMSE	0.5396	0.3385	0.1193	0.0354	0.7022	0.4793	0.1789	0.0679	0.7010	0.5058	0.2132	0.0683
TrajGRU	0.5469	0.4047	0.2695	0.0342	1.2604	1.7699	1.3668	0.0488	0.7071	0.5762	0.4246	0.0661
ExtGAN (Ours)	0.5433	0.4199	0.2351	0.1002	1.0576	1.0232	0.9799	0.7616	0.7041	0.5914	0.3807	0.1821

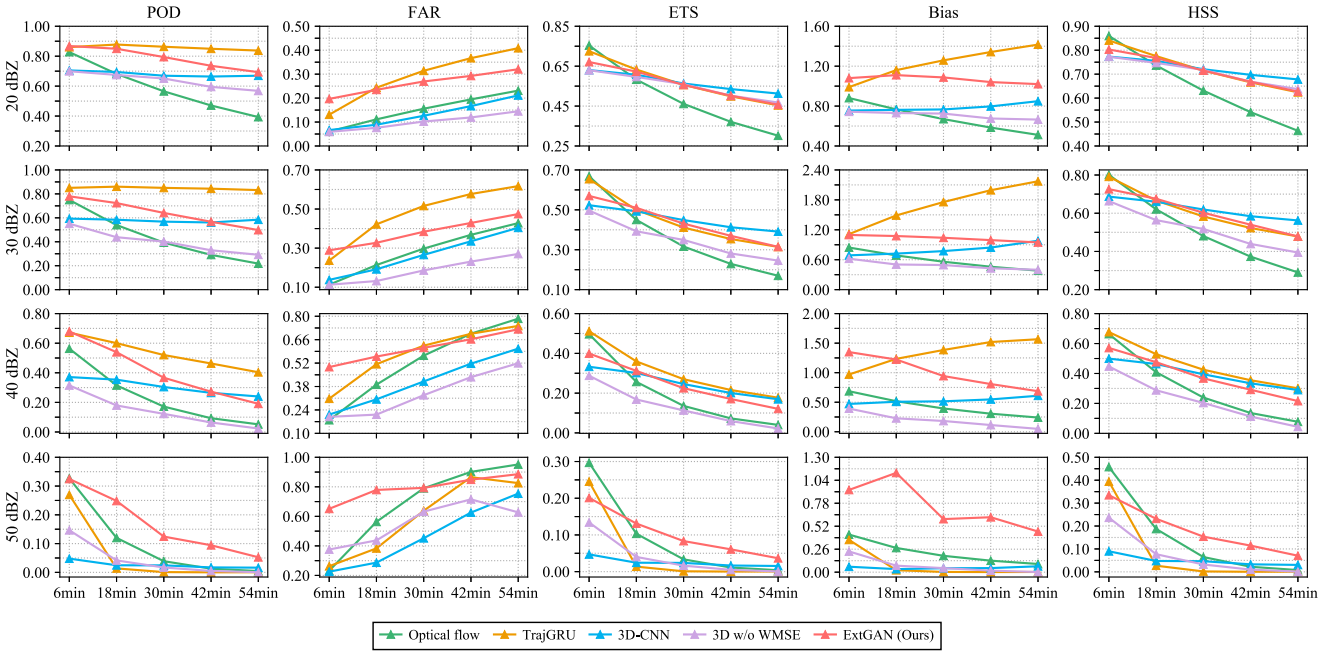

Fig. 6. POD, FAR, ETS, bias, and HSS scores of the models at different forecast lead times. and reflectivity thresholds. Different columns represent different kinds of scores, and different rows have different reflectivity thresholds. The horizontal axis of each subgraph indicates the lead time, and the vertical axis represents the score value.

TABLE III
MR AND AR STATISTICS ON THE TEST SET

Algorithms	Maximum reflectivity (MR)	Average reflectivity (AR)
Label images	48.94±5.56	43.75±2.53
Optical flow	46.28±4.08	42.66±1.78
3D-CNN	46.03±3.90	42.76±1.72
TrajGRU	44.30±3.28	41.59±1.29
ExtGAN (Ours)	49.04±5.71	43.82±2.35

are shown in Fig. 7. Similarly, for each moment, the statistic of label radar images is higher than all extrapolation models, and the MR of ours is closest to labels. The MR of labels hardly changes over time. The gap between the MRs of all extrapolation models and label radar images increases with time.

The ARs of the models also exhibit the same trend as the MRs (see Table II, Fig. 8). The convective cells in label images have the highest AR, and it is almost consistent over time. The AR of ExtGAN is the closest to the labels and also remains stable over time. The ARs of the three baselines have large gaps with labels and decrease rapidly. Three-dimensional-CNN gets the best performance in baseline models.

The consistencies between the number of cells included in the extrapolation results and the number of cells in label images are shown in Fig. 9. For each subgraph, the horizontal axis represents the actual cell number, and the vertical axis indicates the predicted cell number. The subgraphs in the first column are the average results of all test samples. The results of ExtGAN have the best consistency with labels. The results of TrajGRU and 3-D-CNN are similar but slightly worse. The optical flow has the worst results. The second to fifth columns are the results at different times. ExtGAN has the best consistency at every moment. As time increases, the consistencies of all extrapolation models have decreased. TrajGRU and 3-D-CNN perform similarly in this decline. The consistency of the optical flow model declines at the fastest pace, and ExtGAN's consistency declines the slowest.

4) Case Studies: A meteorological process located in Tianjin radar is shown in Fig. 10. The meteorological process's intensity in Fig. 10 is weak, and it is not prone to high-impact meteorological hazards. The convective cells are small and isolated from each other and do not constitute

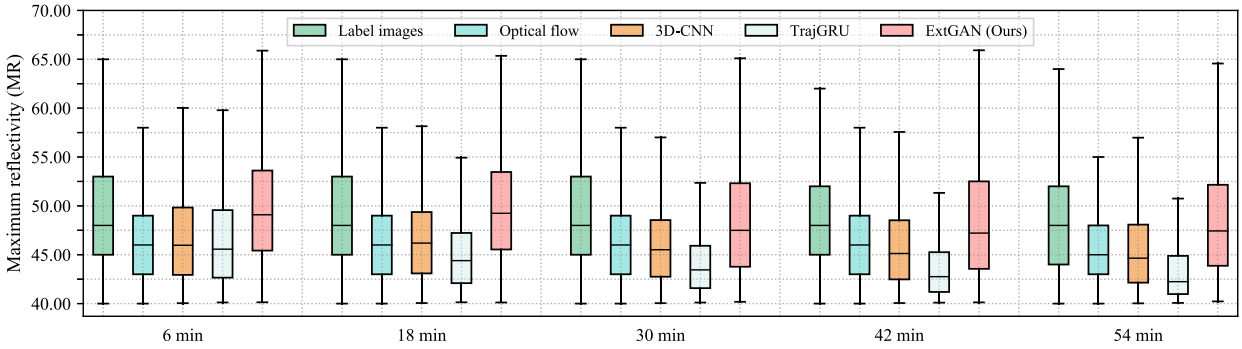

Fig. 7. At different lead times, the maximum reflectivity distribution of convective cells in label images and extrapolation model results.

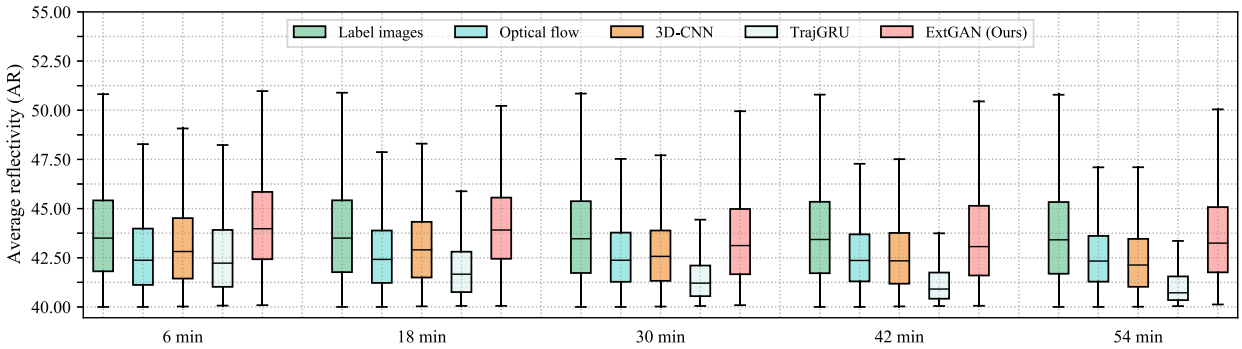

Fig. 8. At different lead times, the average reflectivity distribution of convective cells in label images and extrapolation model results.

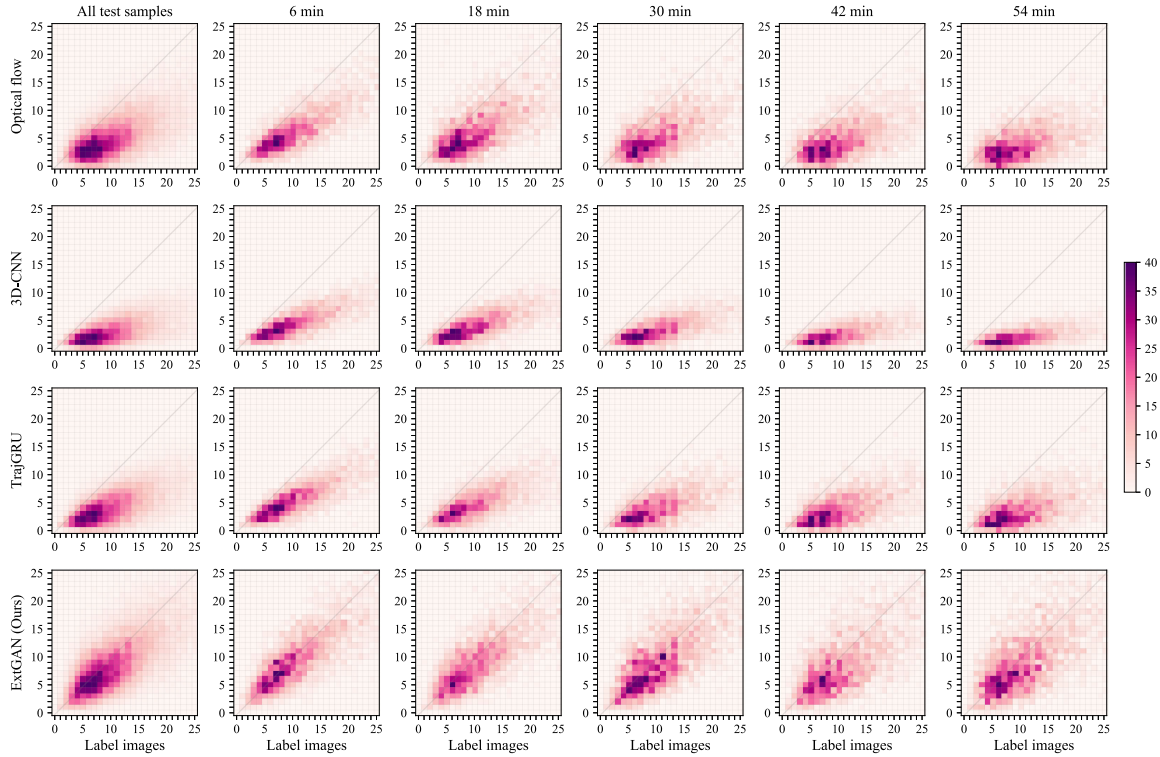

Fig. 9. At different lead times, the consistency of the convective cell number between the extrapolation results and the label images. The first column represents the average of all test samples. The second to fifth columns represent the results at 6, 18, 30, 42, and 54 min, respectively. The horizontal axis of each subgraph represents the number of convective cells contained in the label images, and the vertical axis represents the number of cells included in the extrapolation results.

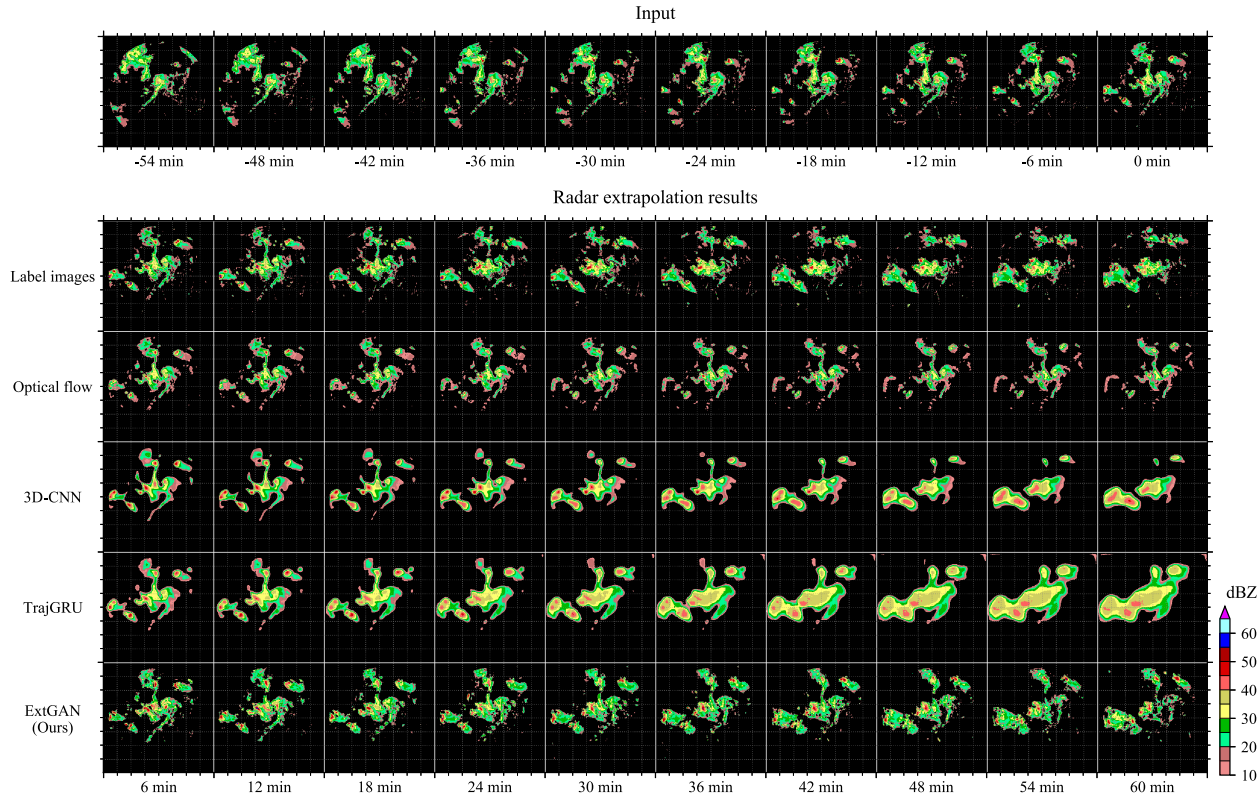

Fig. 10. Extrapolation case located in Tianjin radar at 2016.06.27 07:00-09:00 (UTC). The top row is the input data for radar extrapolation, and the second row is the label images. Each row below is the 1-hour radar extrapolation result of different models.

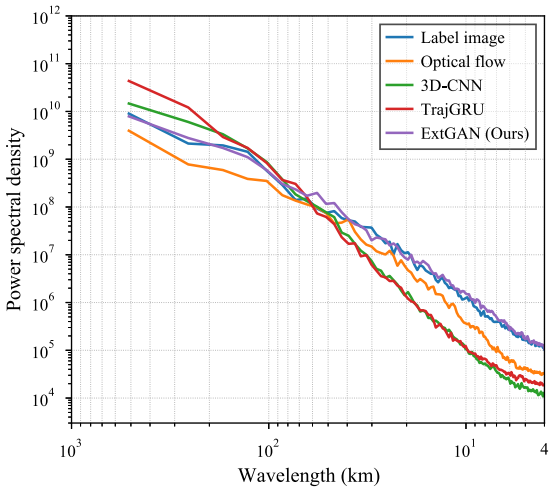

Fig. 11. Radially averaged power spectral density of different models at the last extrapolation moment in Fig. 10.

any convective systems. For extrapolation models, predicting this meteorological process correctly only needs to estimate the motion accurately and consider the intensity variation of the echo but does not need to simulate the nonlinear interaction between cells. All models predict accurately at the first moment. As the lead time increases, the models show significant differences. The optical flow forecasting scale is gradually reduced, and the change of echo intensity is wholly ignored. With time, the echo scales and locations of 3-D-CNN and TrajGRU are approximately the same as the real echo, and the maximum intensity is not significantly weakened. However, there are still differences between these two models. The echo scale extrapolated by 3-D-CNN is small, and the boundaries between the echo regions are clear. TrajGRU tends to exaggerate the forecasting scale and causes each region to stick to each other. These two models also have the same disadvantages. They tend to overestimate the major echo region's intensity but underestimate the strength of convective cells. Simultaneously, the small-scale details in extrapolations are gradually lost, the boundaries become smooth, and the intensity distribution inside the echo cannot be correctly estimated. ExtGAN shows the best performance. The extrapolations of ExtGAN are more realistic, and the distribution and details of each part are also preserved. As time goes by, the quality of our extrapolations does not change significantly, and the intensity is more consistent with the label images. For convective cells in radar images, our extrapolations are closer to real images in position, intensity, and shape details.

A similar result from the energy and wavelength perspective is shown in Fig. 11. Fig. 11 shows the radially averaged power spectral density (RAPSD) [68] of the extrapolated results by different models at the last moment in Fig. 10. RAPSD represents the energy distribution of radar images at different wavelengths. The loss of high-frequency details in echoes of the two ML methods makes them have less energy than the label in the shortwave band. Compared with 3-D-CNN, TrajGRU tends to overestimate the scale of echoes so that it has more

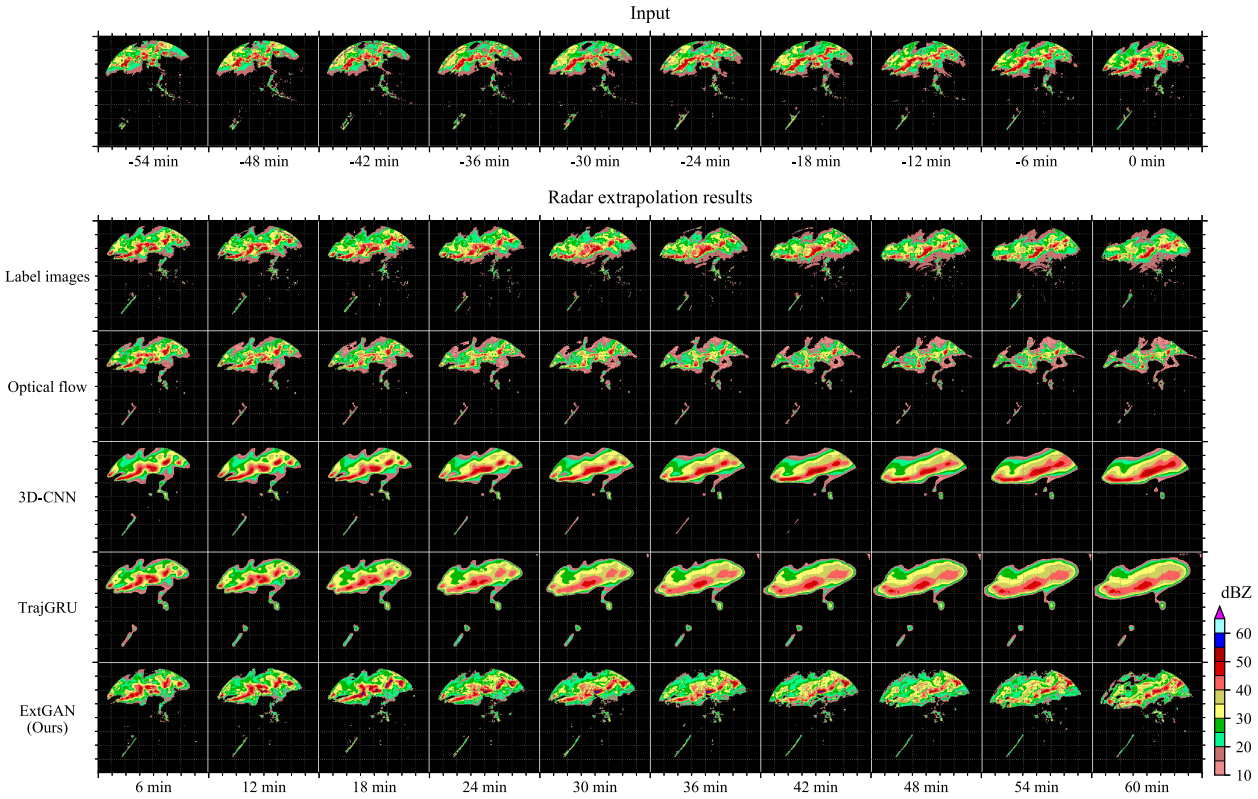

Fig. 12. Extrapolation case located in Tianjin radar at 2015.07.27 11:00-13:00 (UTC). The top row is the input data for radar extrapolation, and the second row is the label images. The third to sixth rows are the 1-hour radar extrapolations of different models.

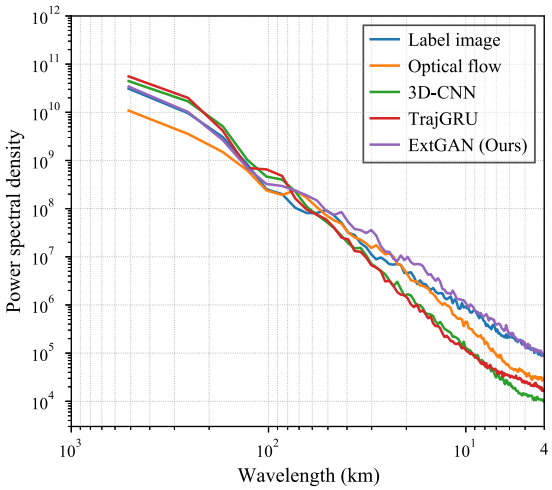

Fig. 13. Radially averaged power spectral density of different models at the last extrapolation moment in Fig. 12.

energy distribution in the longwave band. ExtGAN has a more consistent distribution with the label in the full band, which denotes that our method could make the radar echo have more precise intensity and structure.

A meteorological process, including a convective system, is shown in Fig. 12. The mesoscale convective system composed of multiple convective cells usually causes severe hazards. Correctly predicting this multicell convective system requires models not only to estimate the motion and intensity variations accurately but have the ability to model the complex interaction between convective cells. As shown in Fig. 12, the extrapolation results of all models at the first moment are roughly accurate. With time, the forecasting scale of the optical flow model is reduced. The 3-D-CNN and TrajGRU models are approximately the same as the label images in terms of prediction scale, location, and intensity. Nevertheless, during extrapolation, these two methods gradually lose the details of the echo, and the boundaries of the echo become smooth. Simultaneously, the boundaries between the convective system cells are also lost, and the entire system is predicted to be an intense echo band. The difference is that 3-D-CNN favors predicting a stronger convective system, while TrajGRU tends to predict a larger convective region. ExtGAN still shows the best performance. It not only predicts the movement direction of the convection system accurately but makes the extrapolation result more realistic. More details of the echo are preserved, and the spatial relationship between the cells inside the convective system is more consistent with the label images. Over time, ExtGAN could keep the extrapolation result always high quality without losing details.

The RAPSD of Fig. 12 is shown in Fig. 13. The baseline models lose echo details during the extrapolation process. Compared with the label, their energy in the shortwave band is also lost. Because of underestimating the echo scale, the optical flow method has the smallest energy in the longwave band. ExtGAN could effectively maintain the echo intensity and

structure in the extrapolation result. Therefore, the RAPSDs between the ExtGAN and label are the most consistent.

V. Discussion and Conclusion

A. Pixel-Level Model Performance

The pixel-level evaluation results (see Table II, Fig. 6) show the excellent performance of ExtGAN. Especially for regions greater than 50 dBZ, ExtGAN has significant advantages. These regions indicate cores of convective cells, which usually cause severe hazards. However, ExtGAN's performance is similar to the other ML models for regions with lower reflectance thresholds. Bias score results could explain this phenomenon. As the reflectivity threshold increases, our bias score remains stable, which shows that our model attaches equal importance to the reflectivity of different intensities. However, the TrajGRU and 3-D-CNN methods have higher bias scores at low reflectivity thresholds, and the bias at the high reflectivity threshold are almost zero. This shows that these two models pay more attention to low reflectivity areas during radar extrapolation. In other words, they are more inclined to predict the radar echo into lower intensity. Therefore, these two models have higher scores under lower reflectivity thresholds. However, the price is that TrajGRU and 3-D-CNN almost wholly lack the ability to predict severe hazards. Moreover, the performance of TrajGRU and 3-D-CNN are also different. On the whole, although these two models prefer to predict lower echo intensity, the TrajGRU is more inclined to reduce the error through over-prediction, and the 3-D-CNN tends to reduce the error through under-prediction. Over time (see Fig. 6), the growth rates of TrajGRU's FAR and bias and the attenuation rate of POD and ETS are all greater than that of 3-D-CNN. This indicates that the error accumulation of the TrajGRU model is faster than that of 3-D-CNN. This is caused by the difference between the principles of the two models. TrajGRU uses the same state transition matrix to run time by time during extrapolation, which causes errors to accumulate as the time step increases. When 3-D-CNN performs extrapolation, the extrapolations at all moments are produced simultaneously, which means that the errors of 3-D-CNN do not have such a cumulative process but are more dispersed errors into different lead times. This feature of 3-D-CNN makes the extrapolations' errors at different lead times similar and reduces the impact of latent multimodality on the postprocessing model. Therefore, 3-D-CNN is used for the pre-extrapolation of ExtGAN, and the CGAN is used to overcome the shortcomings of 3-D-CNN and effectively improves ExtGAN's ability to forecast convective hazards.

B. Convective Cell-Based Model Performance

The convective cell-based evaluation confirms the advantages of ExtGAN from another view. The convective cells are the basic units of convective hazard forecasting and many other studies' starting point. Table III shows that ExtGAN's prediction of cell intensity is consistent with the real radar data, while other models' estimations are much lower than the real data. Figs. 7 and 8 further illustrate that ExtGAN has high quality along the time dimension. This result is consistent with the pixel-based evaluation results, indicating that ExtGAN could estimate convective systems more accurately. Fig. 9 shows that ExtGAN also has an excellent ability to predict the number of cells. The cell number could be regarded as an indicator of the separation degree between the cells. The accurate prediction of the cell number could provide a solid foundation for the hazard landing region prediction. The TrajGRU and 3-D-CNN tend to underestimate the number of convective cells. This is due to the disappearance of cell boundaries and the merger of cells caused by underestimating radar echo intensity. ExtGAN overcomes this shortcoming. It benefits from the correct estimation of the cell intensity and the "authenticity" of the extrapolations brought by CGAN. This authenticity has been fully demonstrated in the case studies section (see Figs. 10–12). Compared with other models, our extrapolations not only more visually authentic, but the intensity distribution and internal structure are more in line with the real radar data. Although this does not significantly improve pixel-level scores, it makes the echo has more small-scale details and makes it possible to use the extrapolated results to conduct research based on convective cells.

C. Loss Function

TrajGRU and 3-D-CNN tend to underestimate the radar echo's intensity and cause the loss of details in the echo structure. However, ExtGAN could provide more realistic extrapolations. The difference is that the first two methods use the MSE-based loss function for model training supervision, while we use CGAN to train ExtGAN. The MSE loss function makes the performance of the model worse. This is not only a problem in radar extrapolation but also reported by studies in the ML field [40], [69], [70]. Pixel-level MSE is sensitive to outliers, increases the risk of predicting local extreme values, and causes the model to be more inclined to predict the mean values of the images. When this feature is introduced into the radar extrapolation, the extrapolation model tends to underestimate the echo's intensity, and it also causes the extrapolation model to have a higher bias score at low reflectivity. In addition, MSE also makes high-frequency information in images (such as image texture) disappear and makes images smoother. Reflected into radar extrapolation, the convective cells' boundaries in the extrapolations become smooth and lacks details, and the boundaries between cells become blurry. In this regard, we could cautiously get the inference: when the extrapolations are used to forecast point-by-point, low-intensity, and stable meteorological hazards (e.g., such as stable precipitation, etc.), MSE could be used as the loss function; if the extrapolations are used to forecast severe meteorological hazards based on convective cells (e.g., convective precipitation, hail, etc.), MSE is not suitable as the loss function of the extrapolation model. For the latter, the CGAN used in this article is an excellent method. CGAN can be regarded as a loss function that is adaptively adjusted with model iteration. Furthermore, the CNN structure in CGAN can also effectively characterize the spatial structure of images and avoid the disadvantages caused by pixel-level measurement.

VI. Summary

The contributions of this article are as follows.

1) A two-stage radar extrapolation model named ExtGAN and based on 3-D-CNN and CGAN is proposed. Three-dimensional-CNN is responsible for pre-extrapolation of radar echo, and CGAN is used for postprocessing the results of pre-extrapolation. Finally, accurate and precise radar extrapolation can be obtained.
2) CGAN is used to reduce the blur phenomenon caused by the pixel-based loss function in ML-based models' extrapolations. ExtGAN's extrapolations have more small-scale details and are more consistent with the real radar echo structure.
3) Compared with other ML-based extrapolation models, ExtGAN could more accurately forecast convective cells which usually lead to severe hazards.
4) The applicability of the MSE loss function in the radar extrapolation problem is discussed, and an improved method is proposed.

Although ExtGAN shows excellent performance, some problems need to be solved in the future. The predictability of radar echoes gradually deteriorates over time, because the lifetime of radar echo is finite. Especially for convective systems, when the lead time exceeds the echo lifetime, it is difficult to predict the radar echo in the initial state only based on radar data. We hope to introduce environmental field information into the extrapolation model in the future to improve the prediction of radar echo change and further increase the lead time of radar extrapolation. Moreover, MSE as a pixel-level loss function is not suitable for the extrapolation of convective systems, nor is it suitable for evaluating the extrapolation results from the perspective of convective hazards. CNN can extract high-level features from radar images. These features could characterize radar image information comprehensively from both global structure and local detail. Therefore, the CNN features could be used to replace pixel-level radar reflectivity as the basis for the similarity measurement between radar extrapolations and labels. In the future, we hope to introduce CNN to develop an evaluation method for convective system extrapolation.

Appendix A

TABLE A1
Pre-Extrapolation Model Parameters

Block Name	Kernel Size	In/Out Channels	Output Size
Encoding 1	$3 \times 3 \times 3$	1 / 32	$10 \times 128 \times 128$
Encoding 2	$3 \times 3 \times 3$	32 / 64	$10 \times 64 \times 64$
Encoding 3	$3 \times 3 \times 3$	64 / 128	$10 \times 32 \times 32$
Encoding 4	$3 \times 3 \times 3$	128 / 256	$5 \times 16 \times 16$
$5\times$ Inference	$3 \times 3 \times 3$	256 / 256	$5 \times 16 \times 16$
Decoding 1	$3 \times 3 \times 3$	256 / 128	$10 \times 32 \times 32$
Decoding 2	$3 \times 3 \times 3$	128 / 64	$10 \times 64 \times 64$
Decoding 3	$3 \times 3 \times 3$	64 / 32	$10 \times 128 \times 128$
Decoding 4	$3 \times 3 \times 3$	32 / 32	$10 \times 256 \times 256$
ToImage	$3 \times 3 \times 3$	32 / 1	$10 \times 256 \times 256$

TABLE A2
Postprocessing Model Parameters

Block Name	Kernel Size	In/Out Channels	Output Size
$6\times$ Gen	$3 \times 3 \times 3$	32 / 32	256×256
ToImage	$3 \times 3 \times 3$	32 / 1	256×256
Dis 1	$3 \times 3 \times 3$	33 / 33	256×256
Dis 2	$3 \times 3 \times 3$	33 / 64	128×128
Dis 3	$3 \times 3 \times 3$	64 / 128	128×128
Dis 4	$3 \times 3 \times 3$	128 / 256	64×64
Dis 5	$3 \times 3 \times 3$	256 / 256	32×32
ToScalar	—	256 / 1	1×1

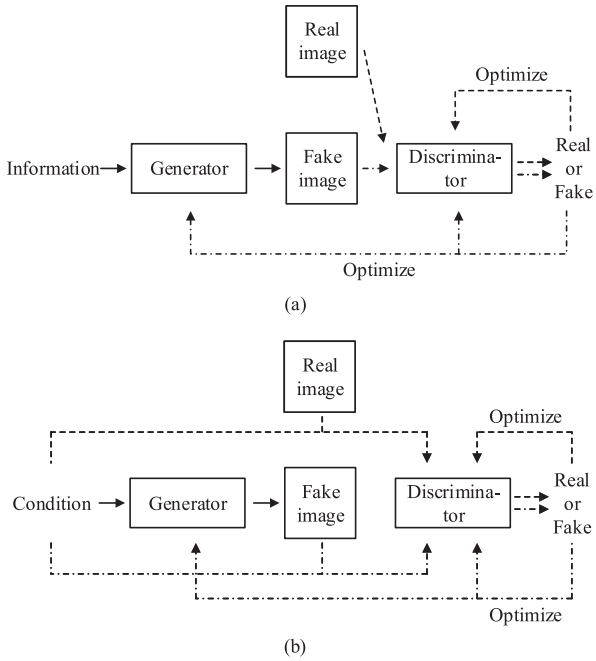

Fig. A1. Optimization process of GAN and CGAN. (a) Optimization process of GAN. The solid line represents the forward propagation of the generator, the dashed line represents the forward propagation and optimization of positive samples, and the dashdotted line represents the forward propagation and optimization of negative samples. (b) Optimization process of CGAN.

References

[1] M. Kunz and P. I. Kugel, "Detection of hail signatures from single-polarization c-band radar reflectivity," *Atmospheric Res.*, vol. 153, pp. 565–577, 2015.
[2] P. Wang, J. Shi, J. Hou, and Y. Hu, "The identification of hail storms in the early stage using time series analysis," *J. Geophysical Res.: Atmospheres*, vol. 123, no. 2, pp. 929–947, 2018.
[3] Y. Yuan, P. Wang, D. Wang, and H. Jia, "An algorithm for automated identification of gust fronts from doppler radar data," *J. Meteorological Res.*, vol. 32, no. 3, pp. 444–455, 2018.
[4] J. Shi, P. Wang, D. Wang, and H. Jia, "Radar-based automatic identification and quantification of weak echo regions for hail nowcasting," *Atmosphere*, vol. 10, no. 6, 2019, Art. no. 325.
[5] B. Medina *et al.*, "A random forest method to forecast downbursts based on dual-polarization radar signatures," *Remote Sens.*, vol. 11, no. 7, 2019, Art. no. 826.
[6] S. Otsuka *et al.*, "Precipitation nowcasting with three-dimensional space-time extrapolation of dense and frequent phased-array weather radar observations," *Weather Forecasting*, vol. 31, no. 1, pp. 329–340, 2016.

[7] Z. Sokol, J. Mejsnar, L. Pop, and V. Bližňák, "Probabilistic precipitation nowcasting based on an extrapolation of radar reflectivity and an ensemble approach," *Atmospheric Res.*, vol. 194, pp. 245–257, 2017.

[8] L. Pop, Z. Sokol, and J. Minářová, "Nowcasting of the probability of accumulated precipitation based on the radar echo extrapolation," *Atmospheric Res.*, vol. 216, pp. 1–10, 2019.

[9] S. Pulkkinen, V. Chandrasekar, and A. Harri, "Fully spectral method for radar-based precipitation nowcasting," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 5, pp. 1369–1382, May 2019.

[10] S. Pulkkinen, V. Chandrasekar, A. von Lerber, and A.-M. Harri, "Nowcasting of convective rainfall using volumetric radar observations," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 7845–7859, Nov. 2020.

[11] M. Dixon and G. Wiener, "Titan: Thunderstorm identification, tracking, analysis, and nowcasting a radar-based methodology," *J. Atmospheric Ocean. Technol.*, vol. 10, no. 10, 1993, Art. no. 785.

[12] J. Johnson *et al.*, "The storm cell identification and tracking algorithm: An enhanced WSR-88 D algorithm," *Weather Forecasting*, vol. 13, no. 2, pp. 263–276, 1998.

[13] L. Han, S. Fu, L. Zhao, Y. Zheng, H. Wang, and Y. Lin, "3D convective storm identification, tracking, and forecasting—An enhanced titan algorithm," *J. Atmospheric Ocean. Technol.*, vol. 26, no. 4, pp. 719–732, 2009.

[14] P. J. Rossi, V. Chandrasekar, V. Hasu, and D. Moisseev, "Kalman filtering-based probabilistic nowcasting of object-oriented tracked convective storms," *J. Atmospheric Ocean. Technol.*, vol. 32, no. 3, pp. 461–477, 2015.

[15] C. Muñoz, L.-P. Wang, and P. Willems, "Enhanced object-based tracking algorithm for convective rain storms and cells," *Atmospheric Res.*, vol. 201, pp. 144–158, 2018.

[16] R. E. Rinehart and E. T. Garvey, "Three-dimensional storm motion detection by conventional weather radar," *Nature*, vol. 273, no. 5660, pp. 287–289, 1978.

[17] L. Li, W. Schmid, and J. Joss, "Nowcasting of motion and growth of precipitation with radar over a complex orography," *J. Appl. Meteorol.*, vol. 34, no. 6, pp. 1286–1300, 1995.

[18] A. Zahraei *et al.*, "Quantitative precipitation nowcasting: A Lagrangian pixel-based approach," *Atmospheric Res.*, vol. 118, pp. 418–434, 2012.

[19] Y. Liu, D.-G. Xi, Z.-L. Li, and Y. Hong, "A new methodology for pixel-quantitative precipitation nowcasting using a pyramid Lucas kanade optical flow approach," *J. Hydrol.*, vol. 529, pp. 354–364, 2015.

[20] H. Zou, S. Wu, J. Shan, and X. Yi, "A method of radar echo extrapolation based on TREC and Barnes filter," *J. Atmospheric Ocean. Technol.*, vol. 36, pp. 1713–1727, 2019.

[21] N. E. Bowler, C. E. Pierce, and A. Seed, "Development of a precipitation nowcasting algorithm based upon optical flow techniques," *J. Hydrol.*, vol. 288, no. 1/2, pp. 74–91, 2004.

[22] W. C. Woo and W. K. Wong, "Operational application of optical flow techniques to radar-based rainfall nowcasting," *Atmosphere*, vol. 2017, no. 8, 2017, Art. no. 48.

[23] M. Diamantakis, "The semi-Lagrangian technique in atmospheric modelling: Current status and future challenges," in *Proc. ECMWF Seminar Numer. Methods Atmos. Ocean Model.*, 2013, pp. 183–200.

[24] U. Germann and I. Zawadzki, "Scale-dependence of the predictability of precipitation from continental radar images. Part I: Description of the methodology," *Monthly Weather Rev.*, vol. 130, no. 12, pp. 2859–2873, 2002.

[25] A. W. Seed, "A dynamic and spatial scaling approach to advection forecasting," *J. Appl. Meteorol.*, vol. 42, no. 3, pp. 381–388, Mar. 2003.

[26] U. Germann and I. Zawadzki, "Scale dependence of the predictability of precipitation from continental radar images. Part II: Probability forecasts," *J. Appl. Meteorol.*, vol. 43, no. 1, pp. 74–89, 2004.

[27] B. J. Turner, I. Zawadzki, and U. Germann, "Predictability of precipitation from continental radar images. Part III: Operational nowcasting implementation (MAPLE)." *J. Appl. Meteorol.*, vol. 43, no. 2, pp. 231–248, Feb. 2004.

[28] N. E. Bowler, C. E. Pierce, and A. W. Seed, "Steps: A probabilistic precipitation forecasting scheme which merges an extrapolation nowcast with downscaled NWP," *Quart. J. Roy. Meteorological Soc.*, vol. 132, no. 620, pp. 2127–2155, 2006.

[29] D. J. Gagne, A. McGovern, and M. Xue, "Machine learning enhancement of storm-scale ensemble probabilistic quantitative precipitation forecasts," *Weather Forecasting*, vol. 29, no. 4, pp. 1024–1043, 2014.

[30] R. Lagerquist, A. McGovern, and T. Smith, "Machine learning for real-time prediction of damaging straight-line convective wind," *Weather Forecasting*, vol. 32, no. 6, pp. 2175–2193, 2017.

[31] Y. Yuan and P. Wang, "Automatic detection of linear mesoscale convective systems," in *Proc. IEEE 13th World Congr. Intell. Control Autom.*, 2018, pp. 170–174.

[32] A. A. Asanjan, T. Yang, K. Hsu, S. Sorooshian, J. Lin, and Q. Peng, "Short-term precipitation forecast based on the Persiann system and LSTM recurrent neural networks," *J. Geophysical Res.: Atmospheres*, vol. 123, no. 22, pp. 12543–12563, 2018.

[33] A. McGovern, C. D. Karstens, T. Smith, and R. Lagerquist, "Quasi-operational testing of real-time storm-longevity prediction via machine learning," *Weather Forecasting*, vol. 34, no. 5, pp. 1437–1451, 2019.

[34] G. E. Jergensen, A. McGovern, R. Lagerquist, and T. Smith, "Classifying convective storms using machine learning," *Weather Forecasting*, vol. 1, pp. 537–559, 2020.

[35] R. Prudden *et al.*, "A review of radar-based nowcasting of precipitation and applicable machine learning techniques," 2020, *arXiv:2005.04988*.

[36] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Conf. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[37] X. Shi *et al.*, "Deep learning for precipitation nowcasting: A benchmark and a new model," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5617–5627.

[38] L. Han, J. Sun, and W. Zhang, "Convolutional neural network for convective storm nowcasting using 3-D doppler weather radar data," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1487–1495, Feb. 2020.

[39] Q.-K. Tran and S.-k. Song, "Multi-channel weather radar echo extrapolation with convolutional recurrent neural networks," *Remote Sens.*, vol. 11, no. 19, 2019, Art. no. 2303.

[40] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *Proc. 4th Int. Conf. Learn. Representations*, 2016, pp. 1–14.

[41] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using lstms," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 843–852.

[42] X. Liang, L. Lee, W. Dai, and E. P. Xing, "Dual motion GAN for future-flow embedded video prediction," in *Proc. IEEE Int. Conf. Comput. Vis. Comput. Soc.*, 2017, pp. 1762–1770.

[43] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," 2018, *arXiv:1804.01523*.

[44] T. Wang *et al.*, "Video-to-video synthesis," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, pp. 1152–1164.

[45] S. Pulkkinen *et al.*, "Pysteps: An open-source python library for probabilistic precipitation nowcasting (v1.0)," *Geoscientific Model Develop.*, vol. 12, no. 10, pp. 4185–4219, 2019.

[46] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[47] J. Jing, Q. Li, X. Ding, N. Sun, R. Tang, and Y. li Cai, "Aenn: A generative adversarial neural network for weather radar echo extrapolation," *ISPRS - Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. XLII-3/W9, pp. 89–94, 2019.

[48] L. Tian, X. Li, Y. Ye, P. Xie, and Y. Li, "A generative adversarial gated recurrent unit model for precipitation nowcasting," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 4, pp. 601–605, Apr. 2020.

[49] P. Xie *et al.*, "An energy-based generative adversarial forecaster for radar echo map extrapolation," *IEEE Geosci. Remote Sens. Lett.*, to be published, doi: 10.1109/LGRS.2020.3023950.

[50] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[51] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proc. IEEE Int. Conf. Comput. Vis., Comput. Soc.*, 2017, pp. 1520–1529.

[52] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

[53] A. Bissoto, E. Valle, and S. Avila, "The six fronts of the generative adversarial networks," 2019, *arXiv:1910.13076*.

[54] X. Huang, Y. Li, O. Poursaeed, J. E. Hopcroft, and S. J. Belongie, "Stacked generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit., IEEE Comput. Soc.*, 2017, pp. 1866–1875.

[55] H. Zhang *et al.*, "StackGAN: Realistic image synthesis with stacked generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, Aug. 2019.

[56] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proc. 6th Int. Conf. Learn. Representations*, 2018, pp. 1–26.

[57] H. Zhang, T. Xu, and H. Li, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis., IEEE Comput. Soc.*, 2017, pp. 5908–5916.

[58] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Proc. 7th Int. Conf. Learn. Representations,* 2019, pp. 1–35.

[59] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,Comput. Vis. Found.*, 2019, pp. 4401–4410.

[60] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[61] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3 d residual networks," in *Proc. IEEE Int. Conf. Comput. Vis., IEEE Comput. Soc.*, 2017, pp. 5534–5542.

[62] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis., IEEE Comput. Soc.*, 2015, pp. 1026–1034.

[64] K. He, X. Zhang, S. Ren, and J. Sun "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit., IEEE Comput. Soc.*, 2016, pp. 770–778.

[65] M. Arjovsky, S. Chintala, and L. Bottou, "GAN Wasserstein," 2017, *arXiv:1701.07875*.

[66] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.

[67] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Proc. Scand. Conf. Image Anal.*,2003, pp. 363–370.

[68] E. Ruzanski and V. Chandrasekar, "Scale filtering for improved nowcasting performance in a high-resolution x-band radar network," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2296–2307, Jun. 2011.

[69] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at signal fidelity measures," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan. 2009.

[70] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 105–114.