DB-RNN: An RNN for Precipitation Nowcasting Deblurring

Zhifeng Ma, Hao Zhang, and Jie Liu, *Fellow, IEEE*

***Abstract*—Precipitation nowcasting based on artificial intelligence has garnered widespread attention in the meteorological and computer communities in recent years. While new models are continuously proposed to refresh the forecasting performance, the problem of gradual blurring of forecast maps as the forecast period extends is still serious. Most models use the mean loss and the recursive prediction structure [such as multiscale recurrent neural network (MS-RNN)]. The mean loss always results in an average of future states, visually appearing as a blur. The recursive prediction method brings the accumulation of error (blur), causing the error (blur) of long-term predictions to increase exponentially. In this study, we add the adversarial loss and gradient loss to penalize the network to ease the blur caused by the averaging loss, and we introduce an additional deblurring network (composed of MS-RNN) behind the forecasting network (composed of MS-RNN) to alleviate the blur caused by the recursive structure, which reduces the blur of the current frame and then recursively and incrementally reduces the blur of subsequent frames. We name the proposed model DB-RNN, which can slow down the error accumulation and alleviate the blurring dilemma. Like MS-RNN, DB-RNN is compatible with multiple recurrent neural network models, such as ConvLSTM, TrajGRU, PredRNN, PredRNN++, MIM, MotionRNN, PrecipLSTM, etc. Experiments on two large radar datasets named HKO-7 and DWD-12 indicate that DB-RNN's predictions are more accurate and clear than those from MS-RNN.**

***Index Terms*—Deblurring, precipitation nowcasting, radar video prediction, recurrent neural network (RNN).**

I. Introduction

PRECIPITATION nowcasting refers to the high-resolution forecasting of precipitation in the short term, with the exact duration definition varying between operational weather centers [1]. It is widely used in traffic and transportation [2], [3], agricultural planting and harvesting [4], [5], marine fishing [6], [7], flight takeoff and landing [8], [9], etc. Precise and timely forecasting will benefit human production and life. If the necessary precautions are taken beforehand, loss of property and life will be avoided. Knowing its importance, it has sparked extensive research and discussion in the meteorological and computer communities [10].

The operational weather and climate forecasting systems have not fundamentally changed for almost 50 years [11]. This type of traditional precipitation forecast, also known as numerical weather prediction, relies on meticulous numerical simulations of physical models to infer the evolution of the atmosphere. Simulation-based systems are often impacted by initial condition fields and require a period of integration to initiate deduction processes. This limitation leads to poor precipitation predictions at 0–2 h lead time [12], [13]. Moreover, such systems are computationally expensive and are unable to provide small-scale forecasting [14]. In contrast, precipitation nowcasting based on radar observations offers an alternative. The optical flow method [15] is the mainstream solution for very short term precipitation forecasting, which extrapolates by calculating the optical flow field. Despite being computationally efficient, this method still struggles with low accuracy in practical applications as it cannot tackle intricate nonlinear transformations of precipitation [16], [17].

Instead of explicitly incorporating prior physical equations, deep learning methods implicitly learn potential meteorological laws by training with massive samples [18], [19]. Although such black box models have been widely criticized for their lack of interpretability, data-driven models have outperformed simulation-based models for both short-term [4], [20], [21] and long-term meteorological prediction tasks [22], [23], [24], [25]. Deep learning models often regard precipitation nowcasting as the radar video prediction issue [26]; they use various networks to encode precipitation movement and decode precipitation prediction [1]. Nevertheless, the earth system is chaotic, high dimensional, and spatiotemporal [27], and it is a particularly challenging task to obtain a robust precipitation representation system. Since introducing a reasonable inductive bias [1] (prior assumption) about the domain knowledge when designing a neural network will reduce the training samples and training difficulty, models [8], [16], [28], [29], [30], [31], [32] integrating a convolutional neural network (CNN) and a recurrent neural network (RNN) stand out among many competitors and are

Manuscript received 22 November 2023; revised 10 January 2024; accepted 9 February 2024. Date of publication 13 February 2024; date of current version 23 February 2024. This work was supported in part by the National Key R&D Program of China under Grant 2021ZD0110900, in part by the National Natural Science Foundation of China under Grant 62106061 and Grant 61972114, in part by the Fundamental Research Funds for the Central Universities under Grant FRFCU5710010521, in part by the Research and Application of Key Technologies for Intelligent Farming Decision Platform, An Open Competition Project of Heilongjiang Province (China) under Grant 2021ZXJ05A03, in part by the Key Research and Development Program of Heilongjiang Province under Grant 2022ZX01A22, and in part by the National Natural Science Foundation of Heilongjiang Province under Grant YQ2019F007. *(Corresponding author: Hao Zhang.)*

Zhifeng Ma and Hao Zhang are with the Internet of Things and Ambient Intelligence Research Center and the National Key Laboratory of Smart Farming Technology and Systems, Harbin Institute of Technology, Harbin 150001, China (e-mail: 20b903027@stu.hit.edu.cn; zhh1000@hit.edu.cn).

Jie Liu is with the National Key Laboratory of Smart Farming Technology and Systems, Harbin Institute of Technology, Harbin 150001, China, and also with the International Research Institute for Artificial Intelligence, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: jieliu@hit.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2024.3365612

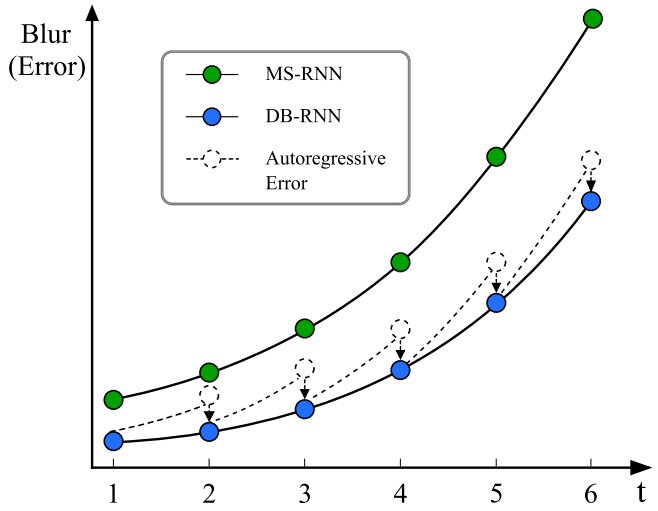

Fig. 1. Mean loss penalizes models for optimizing toward pixel averaging, making predictions of MS-RNN blurry. Autoregression brings accumulation of errors, causing predictions of MS-RNN increasingly blurry. DB-RNN introduces the deblurring model and deblurring loss based on MS-RNN, which makes the prediction of each frame of DB-RNN clearer. Despite the inherent autoregressive error of RNN, the cumulative effect of frame-by-frame deblurring leads to increasingly stronger deblurring capabilities of DB-RNN.

enduring. We call this type of model ConvRNN, which obeys the Markov assumption like RNN and can simultaneously model spatiotemporal patterns.

Radar videos are inherently uncertain and high dimensional. Pixel-by-pixel changes in temporal and spatial dimensions between consecutive frames lead to exponential error accumulation for long-term extrapolation [33] (see Fig. 1). Most models reduce error and deal with uncertainty by averaging forecasts, which leads to increasingly blurred forecasts. In other words, there are two things that cause blur: one is the mean loss (L_1 or L_2) and the other is autoregression (RNN). The former brings about blurry forecasts, while the latter aggravates the phenomenon. In recent years, these ConvRNN models have continued to innovate and break through the forecast bottleneck, which alleviates the blurriness to a certain extent. However, the problem of progressively blurred predictions still haunts most research.

We notice that there is a video deblurring task [34], [35], [36] in the computer vision community, which, like the radar video prediction task, is mainly concerned with the local spatiotemporal dependence of the data. Despite the domain gap, we believe that the ConvRNN networks for precipitation forecasting can also be used for precipitation deblurring. In this article, we propose the deblurring RNN (DB-RNN) to alleviate the autoregression symptom, which consists of two cascaded ConvRNN networks, one for forecasting and one for deblurring. For efficiency, we use a more advanced multiscale version of ConvRNN called multiscale RNN (MS-RNN) [32], which is compatible with many ConvRNNs. We train the two parts sequentially by controlling the corresponding loss weights and finally obtain a powerful end-to-end network for precipitation nowcasting. In addition, we also introduce the gradient difference loss (GDL) [37] and adversarial loss [38] as regularization terms; these constraints narrow the solution space of the neural network, which is beneficial to ease the phenomenon that the mean loss adapts to future uncertainty by making predictions blur. DB-RNN can slow down the blur of each frame. Although there is an inherent cumulative error of RNN, over time, the deblurring effect is accumulated, and the deblurring capability of DB-RNN is even more significant (see Fig. 1). To demonstrate these, we conduct experiments on two huge radar datasets, and the results show that DB-RNN greatly alleviates the blur dilemma of predictions. Our contributions are summarized as follows.

1) We propose a deblurring model named DB-RNN, which consists of two cascaded MS-RNNs.
2) DB-RNN combines the forecasting module (FM) and the deblurring module (DM) and can be trained end to end.
3) DB-RNN is compatible with previous ConvRNN models like MS-RNN, resulting in DB-ConvLSTM, DB-TrajGRU, DB-PredRNN, DB-PredRNN++, DB-MIM, DB-MotionRNN, and DB-PrecipLSTM. We use different basic ConvRNNs to conduct experiments, and the results prove that DB-RNN exceeds MS-RNN in clarity and accuracy.

II. Related Work

A. Radar Video Prediction Models

The majority of studies consider precipitation nowcasting as a radar video prediction problem [39]. They use diverse architectures to encode different inductive biases [1] (prior assumptions) into deep networks. The basic network is either the CNN or the RNN or the generative adversarial network (GAN) [38] or the attention network. The combination of basic networks will constitute a variety of networks. UNet [40] composed of 2-D (RainNet [41], Broad-UNet [42], [43], and SimVP [44]) or 3-D (STConvS2S [5]) convolutions is the simplest precipitation prediction model. ConvRNN, which is composed of convolution (default 2-D) and RNN, is the most effective precipitation prediction model. Their origin is the ConvLSTM [16] proposed by Shi et al. in 2015. Later, a large number of followers continue to refresh the forecasting performance, for example, TrajGRU [8], PredRNN [28], PredRNN++ [45], MIM [29], MotionRNN [31], PrecipLSTM [30], MS-RNN [32], and MS-LSTM [26]. The GAN [17], [19], [46], [47], [48] uses UNet or ConvRNN as the generator and uses one or more discriminators with different roles to play the minimax game. Through the adversarial learning scheme, GAN-based models can get more realistic and accurate extrapolation. The attention network originates from the classic model named Transformer [49] in the natural language processing domain and gradually spreads to the computer vision domain [50]. Recently, Transformer variants have also begun to emerge in the precipitation nowcasting domain, such as Rainformer [51], Earthformer [27], MIMO-VP [52], and LPT-QPN [53]. The research in this article is mainly carried out on the basis of various ConvRNNs because their network designs introduce reasonable spatiotemporal inductive biases.

B. Video Deblurring Models

Generally, video deblurring networks are similar to video prediction networks [34]. Their design all consider the spatiotemporal characteristics of the data and employ a codec structure, where the encoder is used to extract image features and the decoder is used to reconstruct the image. Specifically, part of video deblurring networks [54], [55] uses UNet with 2-D convolutions to learn spatiotemporal dependencies, and they use multiple adjacent frames as input and predict the high-resolution target of the centering frame; part of video deblurring networks [56], [57], [58] directly uses UNet with 3-D convolutions to capture spatial and temporal features, which also use multiple frames to deblur one frame; part of video deblurring networks chooses to use the RNN [59], [60], [61] or the RNN composed of the CNN [62], [63], [64] to capture spatial and temporal changes. In addition to using multiple context frames to deblur the target frame, they also use a single frame as input like ConvRNNs for video prediction since the temporal relation has been implicitly encoded into the recurrent network. Besides, the GAN is also considered to boost them to generate more realistic images that are more in line with human vision [54], [65], [66]. Recently, many Transformer variants [67], [68], [69] have also emerged in the video deblurring domain. It should be pointed out that ConvLSTM is widely used in video deblurring networks [70], [71], but other ConvRNNs (ConvLSTM variants) have not been adopted for video deblurring.

III. Preliminaries

A. Formulation of the Precipitation Nowcasting Problem

In general, given a radar video clip, if we split it into two parts, where the former constitutes the historical sequence, while the latter constitutes the future sequence, then the precipitation nowcasting problem is to predict the maximum possible future sequence given the historical sequence. In detail, if $X_t \in \mathbb{R}^{c\times h\times w}$ is used to represent the radar frame collected at time t, where c, h, and w represent the channel, height, and width, respectively, $X = \{X_0, \ldots, X_{m-1}\}$ is used to represent the historical sequence, and $Y = \{X_m, \ldots, X_{m+n-1}\}$ is used to represent the future sequence, then the problem can be formulated as

$$\hat{Y}^* = \arg\max_{Y} \; P(Y|X) \tag{1}$$

where $\hat{Y}^*$ is the most probable predicted sequence.

B. MS-RNN

Existing ConvRNN models (ConvLSTM [16], TrajGRU [8], PredRNN [28], PredRNN++ [45], MIM [29], MotionRNN [31], and PrecipLSTM [30]) are either getting wider or deeper. Although they gain stronger forecasting capabilities, they will consume huge memory and computing resources. MS-RNN [32] proposes to adopt a multiscale architecture to improve these ConvRNN models, which will make them have less memory and computing requirements but stronger spatiotemporal modeling capabilities while keeping the number of parameters constant. Specifically, MS-RNN reintegrates these increasingly complex tensor flows and embeds the UNet [40] structure into ConvRNN (RNN for short from now on). DB-RNN employs MS-RNN as the basic module; thus, it is compatible with multiple ConvRNN models as MS-RNN, obtaining DB-ConvLSTM, DB-TrajGRU, DB-PredRNN, DB-PredRNN++, DB-MIM, DB-MotionRNN, and DB-PrecipLSTM. This study adopts the superior model MS-PrecipLSTM as a representation to describe DB-RNN. Fig. 2(a) exhibits the architecture of MS-PrecipLSTM from a spatiotemporal or multistep perspective, while Fig. 2(b) exhibits the architecture of MS-PrecipLSTM from a spatial or one-step perspective.

IV. Method

Due to the high dimensionality and inherent uncertainty of natural radar videos, forecasting from models rapidly degrades over time as uncertainty grows, converging to an average of the possible future outcome, visually represented as blurriness [33]. Most models struggle with blurry predictions. The radar video prediction (precipitation nowcasting) problem remains a great challenge.

Given that the computer vision community treats forecasting and deblurring separately, an intuitive idea is to train a deblurring network after training a forecasting network to combat blur [see Fig. 3(a)]. The final prediction can be obtained by inputting the output of the forecasting network to the deblurring network during inference. However, both training and inference of this approach require two steps, and it requires saving the training output of the forecasting network for the training of the deblurring network, which is tedious and time consuming.

In this article, we propose an end-to-end (DB-RNN, which can handle forecasting and deblurring tasks simultaneously [see Fig. 3(b)], which avoids the drawbacks of staged training. DB-RNN consists of two cascaded MS-RNNs with skip connections between their encoders and decoders to fuse same-scale features. DB-RNN has two outputs: one for penalizing the forecasting network and one for penalizing the deblurring network (see Section IV-B). The introduction of the DM results in DB-RNN having the ability to deblur frame by frame. Due to the cumulative effect of autoregression, the deblurring effect of DB-RNN grows over time, which partially makes up for the shortcomings of autoregressive error accumulation in MS-RNN.

During the training process, the purpose of gradually training the two networks is achieved by controlling the weights of the two losses. At the beginning of training, the loss weight of the forecasting network is 1, and the loss weight of the deblurring network is 0.01. At this time, the forecasting network is trained, while the deblurring network is initialization. At the middle of training, the loss weight of the forecasting network is 0.5, and the loss weight of the deblurring network is 0.5. At this time, the forecasting and deblurring networks are trained jointly. At the end of the training, the loss weight of the forecasting network is 0.01, and the loss weight of the deblurring network is 1. At this time, the deblurring network is trained, while the forecasting network is activated to avoid degradation. Since the two networks are trained jointly rather than separately, skip connections are introduced between them. Experiments show

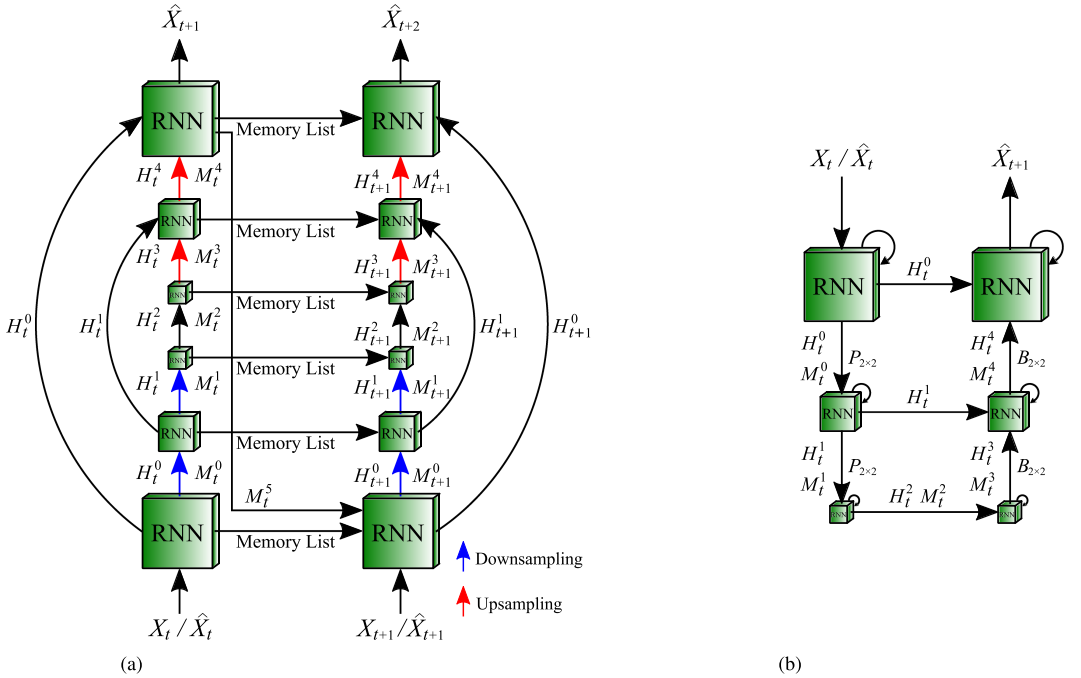

Fig. 2. Architecture of MS-PrecipLSTM [32]. (a) Spatiotemporal or multistep perspective. (b) Spatial or one-step perspective. MS-PrecipLSTM uses six layers (l) of RNN units to form the codec structure. The symbol t in the figure represents time. The hidden state H_t^l and the memory state M_t^l propagate along the vertical (spatial) and zigzag (spatiotemporal) directions, respectively. The hidden states of different scales of the encoder, such as H_t^0 and H_t^1, propagate along the curve (spatial) direction. The memory list propagates along the horizontal (temporal) direction, which is $[H_t^l, C_t^l, \mathrm{MS}_t^l, \mathrm{MT}_t^l]$, where C_t^l, MS_t^l, and MT_t^l denote the cell state, meteorological spatial state, and meteorological temporal state, respectively. Downsampling and upsampling operations are performed by max-pooling ($P_{2\times2}$) and bilinear interpolation ($B_{2\times2}$), respectively.

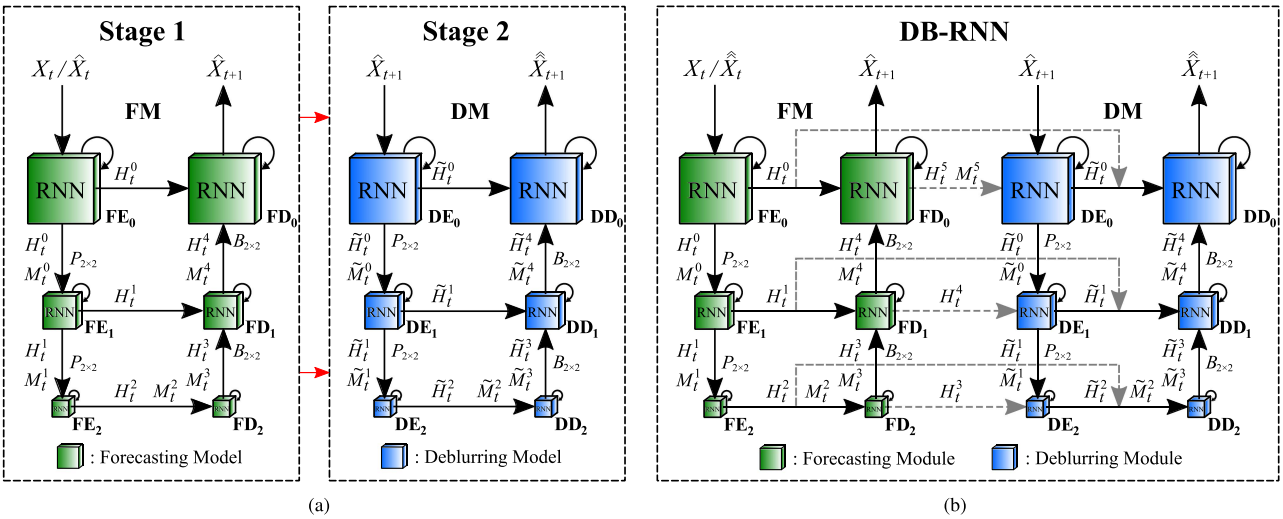

Fig. 3. Two different training strategies for the forecasting task and the deblurring task. (a) Staged training strategy. (b) End-to-end training strategy.

that the end-to-end training strategy of DB-RNN is superior to the staged training strategy (see Section V-G). We will cover the specifics of DB-RNN in the following subsection.

A. DB-RNN

Generally, during one-step prediction, the input of the FM is the true radar map X_t $(0 \le t \le m-1)$ or the deblurred radar map $\hat{\hat{X}}_t$ $(m \le t \le m+n-2)$, and the output of the FM is the forecasted radar map $\hat{X}_{t+1}$. The input of the DM is the forecasted radar map $\hat{X}_{t+1}$, and the output of DM is the deblurred radar map $\hat{\hat{X}}_{t+1}$. Furthermore, we also introduce multiple shortcuts between FM and DM to fuse the same-scale features of the two networks, which can make them interact to jointly combat blurring. Thus, the input of DM also contains the layer output

tensor of FM. These can be formulated as

$$\begin{aligned}\hat{X}_{t+1} &= \mathrm{FM}(X_t/\hat{\hat{X}}_t)\\ \hat{\hat{X}}_{t+1} &= \mathrm{DM}(\hat{X}_{t+1}, (\mathrm{FM}^{\mathrm{out}})_H, (\mathrm{FM}^{\mathrm{out}})_M)\end{aligned} \tag{2}$$

where $(\cdot)^{\mathrm{out}}$ signifies output tensors of "·," while $(\cdot)_H$ and $(\cdot)_M$ signifies the hidden state H_t^l and memory state M_t^l of "·," respectively.

In detail, the forecasting encoder (FE) and the forecasting decoder (FD) of FM all employ three RNN blocks (FE_i and FD_i, $i = 0, 1, 2$). FE adopts the 2×2 max-pooling ($P_{2\times2}$, $(\cdot)^{\downarrow}$) to construct multiscale features. With the stacking of layers, the shape of the radar feature map evolves as $\tilde{c} \times h \times w \to \tilde{c} \times \frac{h}{2} \times \frac{w}{2} \to \tilde{c} \times \frac{h}{4} \times \frac{w}{4}$, where $\tilde{c}$ denotes the embedding dimension. The formulation of FE is

$$\begin{aligned}\mathrm{FE}_0^{\mathrm{out}} &= \mathrm{FE}_0(X_t/\hat{\hat{X}}_t)\\ \mathrm{FE}_1^{\mathrm{out}} &= \mathrm{FE}_1((\mathrm{FE}_0^{\mathrm{out}})^{\downarrow})\\ \mathrm{FE}_2^{\mathrm{out}} &= \mathrm{FE}_2((\mathrm{FE}_1^{\mathrm{out}})^{\downarrow}).\end{aligned} \tag{3}$$

On the contrary, FD uses 2×2 bilinear interpolation ($B_{2\times2}$, $(\cdot)^{\uparrow}$) to construct multiscale features. With the stacking of layers, the shape of the radar feature map evolves as $\tilde{c} \times \frac{w}{4} \times \frac{w}{4} \to \tilde{c} \times \frac{h}{2} \times \frac{w}{2} \to \tilde{c} \times h \times w$. Moreover, FD also accepts skip connections from FE. The formulation of FD is

$$\begin{aligned}\mathrm{FD}_2^{\mathrm{out}} &= \mathrm{FD}_2(\mathrm{FE}_2^{\mathrm{out}})\\ \mathrm{FD}_1^{\mathrm{out}} &= \mathrm{FD}_1(((\mathrm{FD}_2^{\mathrm{out}})_H)^{\uparrow} + (\mathrm{FE}_1^{\mathrm{out}})_H, ((\mathrm{FD}_2^{\mathrm{out}})_M)^{\uparrow})\\ \mathrm{FD}_0^{\mathrm{out}} &= \mathrm{FD}_0(((\mathrm{FD}_1^{\mathrm{out}})_H)^{\uparrow} + (\mathrm{FE}_0^{\mathrm{out}})_H, ((\mathrm{FD}_1^{\mathrm{out}})_M)^{\uparrow})\\ \hat{X}_{t+1} &= (\mathrm{FD}_0^{\mathrm{out}})_H.\end{aligned} \tag{4}$$

In detail, the deblurring encoder (DE) and the deblurring decoder (DD) of DM all employ three RNN blocks too (DE_i and DD_i, $i = 0, 1, 2$). Unlike FE, DE also accepts skip connections from FD, but they are similar overall. According to (3), we can get

$$\begin{aligned}\mathrm{DE}_0^{\mathrm{out}} &= \mathrm{DE}_0(\hat{X}_{t+1} + (\mathrm{FD}_0^{\mathrm{out}})_H, (\mathrm{FD}_0^{\mathrm{out}})_M)\\ \mathrm{DE}_1^{\mathrm{out}} &= \mathrm{DE}_1(((\mathrm{DE}_0^{\mathrm{out}})_{\tilde{H}})^{\downarrow} + (\mathrm{FD}_1^{\mathrm{out}})_H, ((\mathrm{DE}_0^{\mathrm{out}})_{\tilde{M}})^{\downarrow})\\ \mathrm{DE}_2^{\mathrm{out}} &= \mathrm{DE}_2(((\mathrm{DE}_1^{\mathrm{out}})_{\tilde{H}})^{\downarrow} + (\mathrm{FD}_2^{\mathrm{out}})_H, ((\mathrm{DE}_1^{\mathrm{out}})_{\tilde{M}})^{\downarrow})\end{aligned} \tag{5}$$

where $(\cdot)_{\tilde{H}}$ signifies the hidden state $\tilde{H}_t^l$ of "·," while $(\cdot)_{\tilde{M}}$ signifies the memory state $\tilde{M}_t^l$ of "·."

In addition to skip connections from DE, DD also accepts skip connections from FE as input. DD is formulated as

$$\begin{aligned}\mathrm{DD}_2^{\mathrm{out}} &= \mathrm{DD}_2((\mathrm{DE}_2^{\mathrm{out}})_{\tilde{H}} + (\mathrm{FE}_2^{\mathrm{out}})_H, (\mathrm{DE}_2^{\mathrm{out}})_{\tilde{M}})\\ \mathrm{DD}_1^{\mathrm{out}} &= \mathrm{DD}_1(((\mathrm{DD}_2^{\mathrm{out}})_{\tilde{H}})^{\uparrow} + (\mathrm{DE}_1^{\mathrm{out}})_{\tilde{H}} + (\mathrm{FE}_1^{\mathrm{out}})_H,\\ &\qquad ((\mathrm{DD}_2^{\mathrm{out}})_{\tilde{M}})^{\uparrow})\\ \mathrm{DD}_0^{\mathrm{out}} &= \mathrm{DD}_0(((\mathrm{DD}_1^{\mathrm{out}})_{\tilde{H}})^{\uparrow} + (\mathrm{DE}_0^{\mathrm{out}})_{\tilde{H}} + (\mathrm{FE}_0^{\mathrm{out}})_H,\\ &\qquad ((\mathrm{DD}_1^{\mathrm{out}})_{\tilde{M}})^{\uparrow})\\ \hat{\hat{X}}_{t+1} &= (\mathrm{DD}_0^{\mathrm{out}})_{\tilde{H}}.\end{aligned} \tag{6}$$

B. Loss Function

Essentially, two MS-RNNs form the DB-RNN: one for precipitation forecasting and other for precipitation deblurring. Both the subnetworks need to be penalized to perform gradient updates to optimize corresponding parameters. Specifically, both the networks employ the $L_1 + L_2$ pixel loss, which is superior to employ L_1 or L_2 alone [32]. Furthermore, we also add regular terms employing the GDL [37] and adversarial loss [38] to obtain more clear predictions, which partly reduce the mean blur. Finally, the weighted union of these losses results in the overall loss

$$\begin{aligned}L_{\mathrm{for}} &= \frac{\sum_{t=1}^{m+n-1}(|X_t - \hat{X}_t| + |X_t - \hat{X}_t|^2)}{m+n-1}\\ L_{\mathrm{deb}} &= \frac{\sum_{t=1}^{m+n-1}(|X_t - \hat{\hat{X}}_t| + |X_t - \hat{\hat{X}}_t|^2)}{m+n-1}\\ L_{\mathrm{gdl}} &= \frac{\sum_{t=1}^{m+n-1}\sum_{i=0}^{h-2}\sum_{j=0}^{w-2}(||X_t^{i+1,j} - X_t^{i,j}| - |\hat{\hat{X}}_t^{i+1,j} - \hat{\hat{X}}_t^{i,j}|| + ||X_t^{i,j+1} - X_t^{i,j}| - |\hat{\hat{X}}_t^{i,j+1} - \hat{\hat{X}}_t^{i,j}||)}{m+n-1}\\ L_{\mathrm{adv}} &= \frac{\sum_{t=1}^{m+n-1}\mathrm{BCE}(D(\hat{\hat{X}}_t), \mathrm{Real})}{m+n-1}\\ L_{\mathrm{all}} &= \lambda_1 L_{\mathrm{for}} + \lambda_2 L_{\mathrm{deb}} + \lambda_3 L_{\mathrm{gdl}} + \lambda_4 L_{\mathrm{adv}}\end{aligned} \tag{7}$$

where L_{for}, L_{deb}, L_{gdl}, L_{adv}, and L_{all} denote loss of forecasting, loss of deblurring, loss of GDL, loss of adversary, and loss of overall, respectively; λ_1, λ_2, λ_3, and λ_4 are loss weights of L_{for}, L_{deb}, L_{gdl}, and L_{adv} respectively; D is the discriminator; and BCE represents binary cross entropy.

λ_1 decreases linearly from 1 to 0.01 while λ_2 increases linearly from 0.01 to 1 in the first 20 epochs, and λ_1 maintains as 0.01 while λ_2 maintains as 1 in the remaining training epochs. The reason is that we want to start with mainly training the forecasting network ($\lambda_1 = 1$) while appropriately activating the deblurring network ($\lambda_2 = 0.01$), then gradually transition to mainly training the deblurring network ($\lambda_2 = 1$) while avoiding completely degradation of the forecasting network ($\lambda_1 = 0.01$), and finally train both networks to work together. λ_3 and λ_4 remain unchanged throughout the training process, which is set to 0.001 and 1, respectively. L_{adv} is made of the BCE loss function, and we want to trick the discriminator (D) into recognizing the deblurred frame ($\hat{\hat{X}}_t$) generated by the generator (DB-RNN) as real (1), at which L_{adv} takes the minimum value (0). Like the generator, the parameters of the discriminator are updated during each mini-batch gradient descent process, but with a separate optimizer and loss function

$$L_D = \mathrm{BCE}(D(X_t), \mathrm{Real}) + \mathrm{BCE}(D(\hat{\hat{X}}_t), \mathrm{Fake}) \tag{8}$$

where fake is equal to 0. The discriminator is trained to distinguish real from fake ($L_D \downarrow$, $L_{\mathrm{all}} \uparrow$), which forces the generator to

TABLE I
DETAILS OF DISCRIMINATORS

Encoders × Blocks	Decoders × Blocks	Models × Params
[Conv2d, LeakyReLU, MaxPool2d] × 3	[Linear, LeakyReLU, Linear, Sigmoid] × 1	[DB-ConvLSTM, DB-TrajGRU] × 0.012M
		[DB-PredRNN, DB-PredRNN++] × 0.017M
		[DB-MIM, DB-MotionRNN, DB-PrecipLSTM] × 0.027M

TABLE II
PARAMETER SETTINGS OF LAYERS IN TABLE I

Layers	in_c	out_c	k	s	p
Conv2d	1/1/1	16/16/16	3	1	1
MaxPool2d	16/16/16	16/16/16	4	4	0
Conv2d	16/16/16	20/20/32	3	1	1
MaxPool2d	20/20/32	20/20/32	4	4	0
Conv2d	20/20/32	20/32/32	3	1	1
MaxPool2d	20/32/32	20/32/32	2	2	0
Linear	500/800/800	10/10/16	1	1	0
Linear	10/10/16	1/1/1	1	1	0

in_c, out_c, k, s, and p are abbreviations of the input channel, output channel, kernel, stride, and padding, respectively.

generate more realistic frames ($L_D \uparrow$, $L_{\text{all}} \downarrow$), which then causes the discriminator to update ($L_D \downarrow$, $L_{\text{all}} \uparrow$), and so on. Through the adversarial training of ebb and flow, the two reach the Nash equilibrium [72], and the training ends.

The discriminator is composed of a convolutional encoder and a linear decoder (see Table I). Given that base RNNs used by DB-RNN have different parameters [32], that is, generators have different parameters, we equip models with three discriminators with different parameters, 0.012M for DB-ConvLSTM and DB-TrajGRU, 0.017M for DB-PredRNN and DB-PredRNN++, and 0.027M for DB-MIM, DB-MotionRNN, and DB-PrecipLSTM, which is achieved by changing input or output channels of layers (see Table II).

V. EXPERIMENTS

A. Datasets

We conduct experiments with the HKO-7 [8] and DWD-12 [41] datasets, where the former is collected by one Doppler radar, while the latter is collected by 17 Doppler radars. HKO-7 has seven years of data from 2009 to 2015, covering an area of 512×512 km^2 centered in Hong Kong. The temporal resolution of the dataset is 6 min, and the spatial resolution is 1.07 km. We only employ data from rainy days, that is, 812 days for training, 50 days for validation, and 131 days for testing. DWD-12 has 12 years of data from 2006 to 2017, covering an area of 900×900 km^2 across the entire Germany. The temporal resolution of the dataset is 5 min, and the spatial resolution is 1 km. We use data from 2006 to 2013 for training, data from 2014 to 2015 for validation, and data from 2016 to 2017 for testing.

The conversion relationship between the image pixel value (P: 0–255) and the radar reflectivity intensity (dBZ: 0–60) for both the datasets is

$$P = \left\lfloor 255 \times \frac{\text{dBZ}}{60} \right\rfloor. \tag{9}$$

The conversion relations between the radar reflectivity intensity (dBZ: 0–60) and rainfall intensity (R: mm/h) for the HKO-7 and DWD-12 datasets are

$$\begin{aligned} \text{dBZ} &= 10 \times \lg\left(58.53 \times R^{1.56}\right) \\ \text{dBZ} &= 10 \times \lg\left(256 \times R^{1.42}\right) \end{aligned} \tag{10}$$

respectively.

B. Implementation Details

We apply the same experimental setup to all models. The kernel size and hidden channel of RNN are 3×3 and 24, respectively. MS-RNN is stacked with six layers of RNN. We use the Adam optimizer [73] with an initial learning rate of 0.0003. The batch size is 4. We interpolate radar maps to 160×160 for quick verification. We sample frames every two frames for HKO-7 and five frames for DWD-12, which increases the difficulty of nowcasting. The task of the model is to predict five future frames based on five historical frames. We train MS-RNN for 25 epochs and DB-RNN for 50 epochs using the NVIDIA A100 GPU. All leaky ReLU layers in discriminators use a negative slope of 0.01. There is nothing to set up for sigmoid layers. The specific parameter settings of other layers in Table I are shown in Table II.

C. Metrics

We evaluate models using the critical success index (CSI) [8] and the Heidke skill score (HSS) [8] at different rainfall intensity thresholds (0.5, 2, 5, 10, and 30 mm/h). First, we convert pixel values of predicted or real images to 0 or 1 by the threshold τ. In detail, we employ (9) and (10) to transform the pixel value (P) to rainfall R. If $R \geq \tau$, P will be 1; otherwise, P will be 0. Second, we can count TP (prediction = 1, truth = 1), FN (prediction = 0, truth = 1), FP (prediction = 1, truth = 0), and TN (prediction = 0, truth = 0), separately. Finally, CSI, HSS, and POD are calculated as $(\text{TP}/(\text{TP}+\text{FN}+\text{FP}))$, $((\text{TP}\times\text{TN}-\text{FN}\times\text{FP})/((\text{TP}+\text{FN})(\text{FN}+\text{TN})+(\text{TP}+\text{FP})(\text{FP}+\text{TN})))$, and $(\text{TP}/(\text{TP}+\text{FN}))$ respectively. In addition, the balanced mean squared error (B-MSE) [8] and balanced mean absolute error (B-MAE) [8] are also used to eliminate the interference of the long tail distribution of precipitation to focus on heavy rains.

D. Quantitative Experiments

We use seven basic RNN models to quantitatively compare the performance of MS-RNN and DB-RNN, which are ConvLSTM [16], TrajGRU [8], PredRNN [28], PredRNN++ [45], MIM [29], MotionRNN [31], and PrecipLSTM [30]. It should be pointed out that Hong Kong has a subtropical monsoon climate with different levels of precipitation, while Germany has a temperate maritime climate with mostly light rain and less heavy rain. Therefore, thresholds 0.5, 2, 5, 10, and 30 mm/h are used when measuring HKO-7, while only thresholds 0.5, 2, and

TABLE III
QUANTITATIVE COMPARISON ON HKO-7

Models	CSI ↑					HSS ↑					B-MSE ↓	B-MAE ↓
	$R \geq 0.5$	$R \geq 2$	$R \geq 5$	$R \geq 10$	$R \geq 30$	$R \geq 0.5$	$R \geq 2$	$R \geq 5$	$R \geq 10$	$R \geq 30$		
MS-ConvLSTM [32]	0.668	0.615	0.519	0.381	0.201	0.785	0.748	0.669	0.531	0.309	268.8	971.7
DB-ConvLSTM	**0.679**	**0.626**	**0.536**	**0.406**	**0.243**	**0.794**	**0.757**	**0.684**	**0.557**	**0.367**	**252.0**	**926.3**
MS-TrajGRU [32]	0.669	0.617	0.523	0.390	0.205	0.787	0.750	0.674	0.543	0.316	258.1	961.5
DB-TrajGRU	**0.683**	**0.633**	**0.543**	**0.418**	**0.233**	**0.797**	**0.763**	**0.691**	**0.574**	**0.358**	**238.7**	**912.1**
MS-PredRNN [32]	0.677	0.623	0.534	0.408	0.236	0.792	0.754	0.682	0.558	0.356	247.4	925.5
DB-PredRNN	**0.687**	**0.640**	**0.557**	**0.438**	**0.272**	**0.799**	**0.768**	**0.703**	**0.594**	**0.407**	**237.4**	**882.0**
MS-PredRNN++ [32]	0.678	0.628	0.543	0.418	0.243	0.793	0.758	0.690	0.572	0.367	244.7	913.6
DB-PredRNN++	**0.686**	**0.638**	**0.553**	**0.429**	**0.260**	**0.798**	**0.766**	**0.700**	**0.583**	**0.391**	**241.4**	**891.7**
MS-MIM [32]	0.676	0.624	0.536	0.408	0.233	0.792	0.755	0.684	0.560	0.352	250.2	927.9
DB-MIM	**0.684**	**0.636**	**0.553**	**0.439**	**0.282**	**0.797**	**0.764**	**0.700**	**0.595**	**0.421**	**241.5**	**891.0**
MS-MotionRNN [32]	0.680	0.629	0.540	0.411	0.232	0.794	0.760	0.688	0.564	0.353	246.5	918.0
DB-MotionRNN	**0.692**	**0.644**	**0.559**	**0.440**	**0.270**	**0.803**	**0.771**	**0.705**	**0.596**	**0.403**	**226.6**	**863.5**
MS-PrecipLSTM [32]	0.685	0.637	0.552	0.430	0.260	0.798	0.766	0.698	0.583	0.385	232.1	884.2
DB-PrecipLSTM	**0.691**	**0.644**	**0.562**	**0.448**	**0.285**	**0.802**	**0.771**	**0.708**	**0.604**	**0.424**	**230.7**	**862.3**

The darker the better. "$R \geq \tau$" means the nowcasting skill score at threshold τ mm/h.
The boldface indicates the optimal metric value in the current subtable, and each subtable indicates the comparison between MS-RNN and DB-RNN.

TABLE IV
QUANTITATIVE COMPARISON ON DWD-12

Models	CSI ↑			HSS ↑			POD ↑			B-MSE ↓	B-MAE ↓
	$R \geq 0.5$	$R \geq 2$	$R \geq 5$	$R \geq 0.5$	$R \geq 2$	$R \geq 5$	$R \geq 0.5$	$R \geq 2$	$R \geq 5$		
MS-ConvLSTM [32]	0.437	0.275	0.135	0.592	0.414	0.225	0.512	0.326	0.158	159.7	478.6
DB-ConvLSTM	**0.459**	**0.304**	**0.151**	**0.615**	**0.453**	**0.250**	**0.547**	**0.365**	**0.177**	**153.1**	**461.9**
MS-TrajGRU [32]	0.416	0.241	0.104	0.570	0.371	0.178	0.478	0.274	0.116	164.7	492.2
DB-TrajGRU	**0.451**	**0.302**	**0.175**	**0.607**	**0.451**	**0.288**	**0.535**	**0.367**	**0.212**	**156.1**	**468.6**
MS-PredRNN [32]	0.451	0.271	0.100	0.607	0.411	0.171	0.524	0.306	0.108	**149.7**	462.0
DB-PredRNN	**0.459**	**0.295**	**0.133**	**0.614**	**0.442**	**0.224**	**0.535**	**0.341**	**0.150**	152.4	**455.3**
MS-PredRNN++ [32]	0.456	0.287	0.117	0.611	0.430	0.198	0.532	0.332	0.129	**149.4**	457.4
DB-PredRNN++	**0.466**	**0.317**	**0.160**	**0.621**	**0.468**	**0.265**	**0.550**	**0.378**	**0.185**	151.2	**451.6**
MS-MIM [32]	0.459	0.286	0.102	0.614	0.430	0.175	0.538	0.328	0.110	**149.3**	458.6
DB-MIM	**0.467**	**0.306**	**0.156**	**0.621**	**0.453**	**0.258**	**0.552**	**0.362**	**0.180**	153.6	**453.7**
MS-MotionRNN [32]	0.464	0.293	0.127	0.619	0.437	0.213	0.548	0.341	0.143	**147.0**	459.6
DB-MotionRNN	**0.466**	**0.307**	**0.158**	**0.621**	**0.457**	**0.264**	**0.546**	**0.358**	**0.179**	149.5	**451.6**
MS-PrecipLSTM [32]	0.462	0.296	0.147	0.617	0.439	0.242	0.541	0.349	0.171	**146.0**	455.2
DB-PrecipLSTM	**0.476**	**0.320**	**0.174**	**0.630**	**0.471**	**0.284**	**0.571**	**0.385**	**0.208**	148.5	**446.9**

The darker the better. "$R \geq \tau$" means the nowcasting skill score at threshold τ mm/h.
The boldface indicates the optimal metric value in the current subtable, and each subtable indicates the comparison between MS-RNN and DB-RNN.

5 mm/h are used when measuring DWD-12. Tables III and IV show quantitative comparison experiments on the HKO-7 and DWD-12 datasets, respectively.

Overall, all DB-RNNs outperform MS-RNNs on all metrics on the HKO-7 dataset, and all DB-RNNs outperform MS-RNNs on all metrics on the DWD-12 dataset except for the B-MSE indicator. Specifically, we have the following.

1) For the HKO-7 dataset, compared with MS-ConvLSTM, the CSI-0.5, CSI-2, CSI-5, CSI-10, and CSI-30 of DB-ConvLSTM increase by 1.6%, 1.8%, 3.3%, 6.6%, and 20.9%, respectively, while the B-MSE and B-MAE of DB-ConvLSTM decrease by 6.3% and 4.7%, respectively.
2) For the DWD-12 dataset, compared with MS-PrecipLSTM, the HSS-0.5, HSS-2, and HSS-5 of DB-PrecipLSTM increase by 2.1%, 7.3%, and 17.4%, respectively, while the B-MAE of DB-PrecipLSTM decreases by 1.8%. Obviously, we can get the same conclusion as on HKO-7: the greater the precipitation level, the more the DB-RNN improves. In other words, DB-RNN cares more about heavier rain, which is more harmful.
3) Fig. 4 also proves this point. Fig. 4 exhibits the framewise CSI and HSS metrics at different thresholds on the HKO-7 dataset. Over time (horizontal axis), the performance of both MS-RNN and DB-RNN declines significantly, that is, it becomes increasingly blurry. However, DB-RNN alleviates this tendency, thanks to the introduction of the deblurring network and deblurring loss. In addition, it can also be seen from Fig. 4 that the deblurring effect of DB-RNN increases with time steps, especially for higher levels

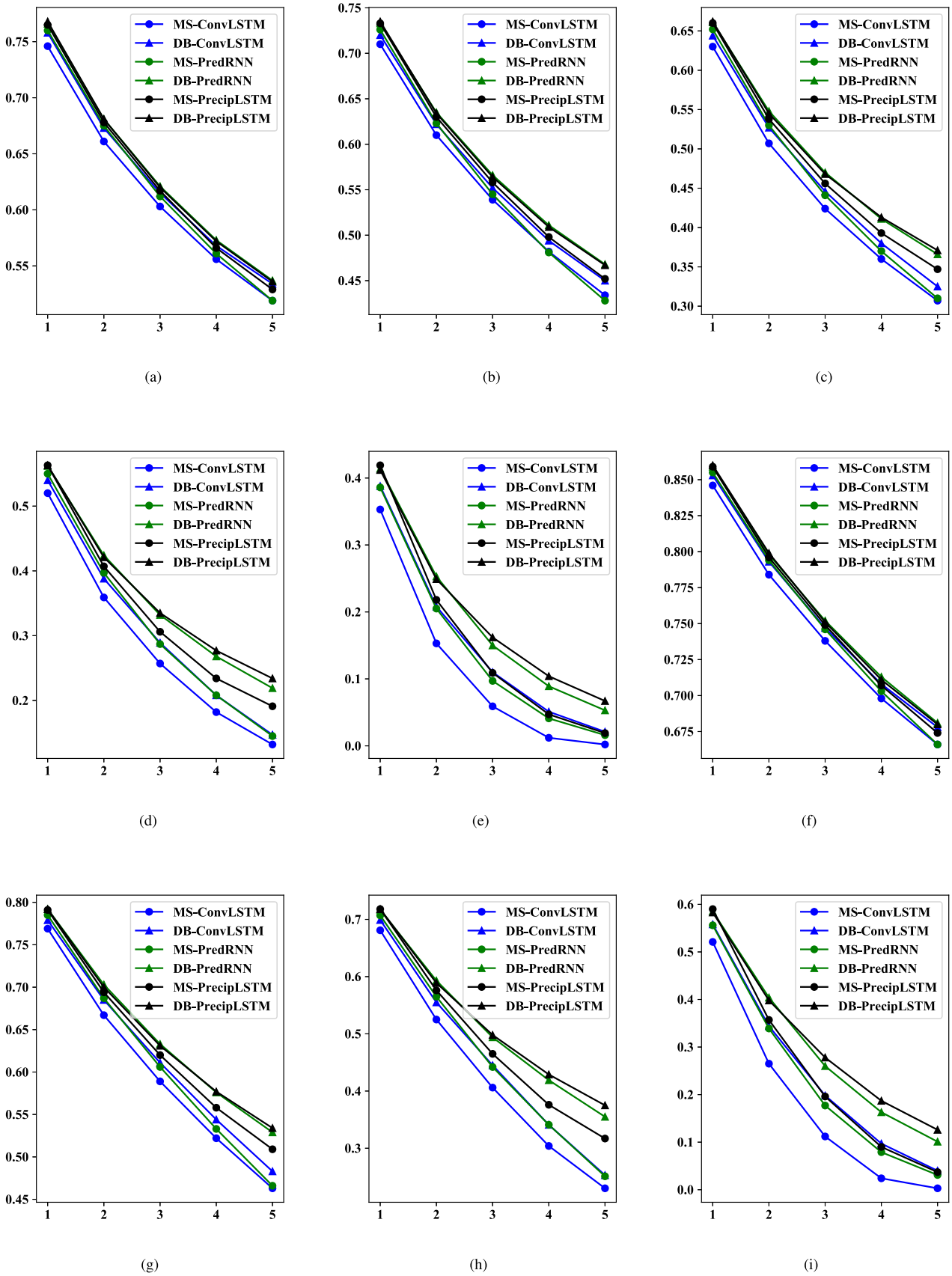

Fig. 4. Comparison of framewise CSI and HSS at different rainfall thresholds on HKO-7. CSI-τ and HSS-τ are shorts for CSI ($R \geq \tau$) and HSS ($R \geq \tau$), respectively, which mean nowcasting skill scores at threshold τ mm/h. For layout convenience, we omit the HSS-2 indicator. (a) CSI-0.5. (b) CSI-2. (c) CSI-5. (d) CSI-10. (e) CSI-30. (f) HSS-0.5. (g) HSS-5. (h) HSS-10. (i) HSS-30.

of precipitation, which echoes Fig. 1, that is, deblurring is accumulated.

E. Qualitative Experiments

Qualitative experiments are similar to quantitative experiments. Seven basic RNN models are also used to compare MS-RNN and DB-RNN, namely, ConvLSTM [16], TrajGRU [8], PredRNN [28], PredRNN++ [45], MIM [29], MotionRNN [31], and PrecipLSTM [30]. We select a total of six precipitation cases to compare them: two from the HKO-7 dataset (see Fig. 5) and four from the DWD-12 dataset (see Figs. 6 and 7).

Overall, DB-RNN's predictions are clearer and more accurate than MS-RNN's, and DB-RNN pays more attention to heavier rain than MS-RNN. Furthermore, the gap between DB-RNN and MS-RNN is not large at the beginning, and the predictions of both are not very vague. However, as time goes by, the gap between the two begins to appear, especially the prediction of the last frame, that is, the gradual blurring trend of DB-RNN is not as fast as that of MS-RNN. This again demonstrates the deblurring accumulation ability of DB-RNN, echoing Figs. 1 and 4.

Specifically, we have the following.

1) Fig. 5(a) shows a process of sparse precipitation moving eastward, and Fig. 5(b) shows a process of dense precipitation moving eastward. By comparing the last frame ($t = 9$) predicted by MS-RNN and DB-RNN, we can conclude that DB-RNN has the ability to deblur, predict more accurately, and pay more attention to heavy rain (yellow and red pixels).
2) Fig. 6(a) shows a process of increasing precipitation moving northward, and Fig. 6(b) shows a process of decreasing precipitation moving eastward. The case in Fig. 6(a) is extremely difficult. The model needs to infer changes for the unknown five frames ($t = 5, \ldots, 9$) from the known two frames ($t = 3, 4$) with significant precipitation. MS-ConvLSTM, MS-TrajGRU, and DB-PredRNN++ believe that precipitation will dissipate, DB-TrajGRU and DB-PredRNN believe that precipitation will maintain, while DB-ConvLSTM, MS-PredRNN, MS-PredRNN++, MS-MIM, DB-MIM, MS-MotionRNN, DB-MotionRNN, MS-PrecipLSTM, and DB-PrecipLSTM believe that precipitation will expand. In general, the predictions of most DB-RNNs are clearer than MS-RNNs', and most DB-RNNs believe that precipitation will remain or expand, among which the prediction of DB-PrecipLSTM is the clearest and most accurate. In comparison, the example in Fig. 6(b) is much simpler. The model needs to infer changes for the unknown five frames ($t = 5, \ldots, 9$) from the known five frames ($t = 0, \ldots, 4$) with significant precipitation. DB-RNN still outperforms MS-RNN in terms of clarity and accuracy, indicating that DB-RNN can capture not only the birth but also the death of precipitation;
3) Fig. 7(a) shows a process of the narrow precipitation band moving eastward, and Fig. 7(b) shows a process of the broad precipitation cluster spreading eastward. By comparing the last frame ($t = 9$) predicted by MS-RNN and DB-RNN, we can get the same conclusion as before: DB-RNN resists blur, predicts more accurately, and pays more attention to heavier rain (yellow pixels).

TABLE V
COMPLEXITIES OF MS-RNN AND DB-RNN ON HKO-7

Models	Params	FLOPs	Memory	Time
MS-ConvLSTM	0.25 M	25.2 G	0.96 G	1.12 h
DB-ConvLSTM	0.50 M	50.3 G	1.50 G	4.74 h
MS-PredRNN	0.51 M	51.0 G	1.39 G	4.08 h
DB-PredRNN	1.01 M	102.0 G	2.29 G	15.40 h
MS-MIM	0.99 M	80.2 G	2.00 G	4.74 h
DB-MIM	1.99 M	160.4 G	3.39 G	18.83 h

F. Analysis of Complexity and Scalability

We choose ConvLSTM, PredRNN, and MIM as the basic RNNs to compare the various complexities of MS-RNN and DB-RNN. Table V shows the comparison on HKO-7. The parameter complexity, computational complexity (FLOPs), space complexity (memory), and time complexity of DB-RNN are approximately two times, two times, two times, and four times that of MS-RNN, respectively. DB-RNN uses two MS-RNNs in sequence, and these increases are not surprising. Although DB-RNN consumes more computing resources, the performance of DB-RNN is stronger than MS-RNN with almost the same parameters, such as DB-ConvLSTM is stronger than MS-PredRNN, and DB-PredRNN is stronger than MS-MIM (see Tables III and IV). The most fatal problem when training neural networks is video memory. Improperly designed neural networks may lead to the exhaustion of video memory. Thanks to the multiscale design of MS-RNN, DB-RNN takes up less video memory and is still within the tolerance of normal video memory. Finally, the training time of DB-RNN has increased excessively, which should be the result of the introduction of adversarial training. Although the training time is extended, the benefits in exchange are stronger performance and reduced blur effects.

This article only uses seven basic RNN models to verify the advantages of DB-RNN, namely, ConvLSTM [16], TrajGRU [8], PredRNN [28], PredRNN++ [45], MIM [29], MotionRNN [31], and PrecipLSTM [30]. In practice, there are many variants of ConvLSTM. Thanks to the scalability and compatibility of MS-RNN, DB-RNN is compatible with most convolutional RNN models, such as SA-ConvLSTM [74], MoDeRNN [75], CMS-LSTM [76], PredRNN-V2 [77], MK-LSTM [26], etc. Due to space limitations, this article no longer performs corresponding experiments.

G. Comparison With Staged Training Models

We use ConvLSTM [16] and PredRNN [28] as base models to compare the performance of MS-RNN [32], staged training MS-RNN (MS-RNN-S), and DB-RNN. The experimental results are shown in Table VI. Overall, DB-RNN outperforms MS-RNN-S, while MS-RNN-S outperforms MS-RNN. We can conclude that

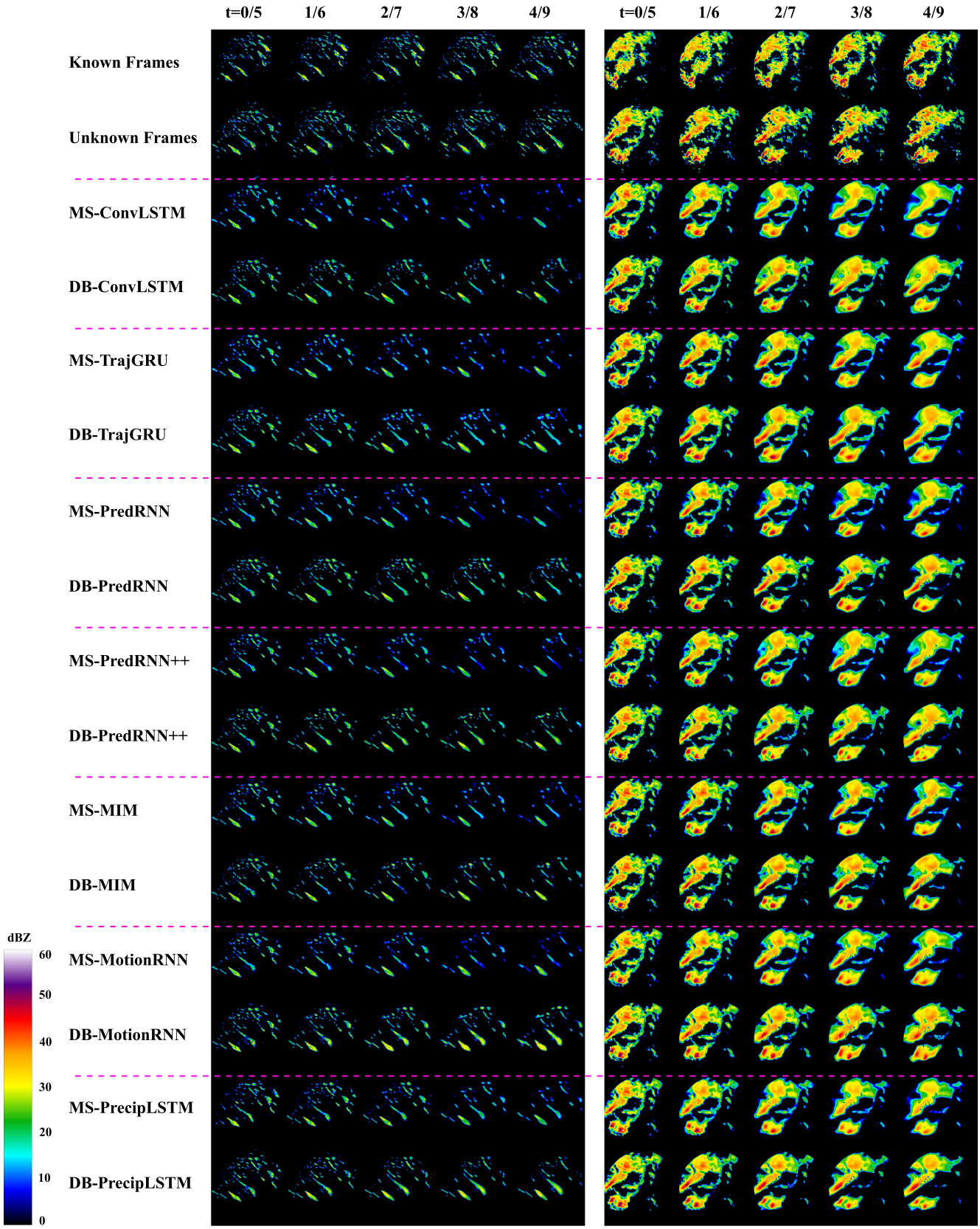

Fig. 5. Cases study of instantaneous precipitation starting at 2012-01-24 17:54:00 UTC+8 (left) and 2013-03-30 06:54:00 UTC+8 (right) of Hong Kong. The first row is known historical frames, the second row is unknown future frames, and the other rows are predicted frames. (a) Process of sparse precipitation moving eastward. (b) Process of dense precipitation moving eastward.

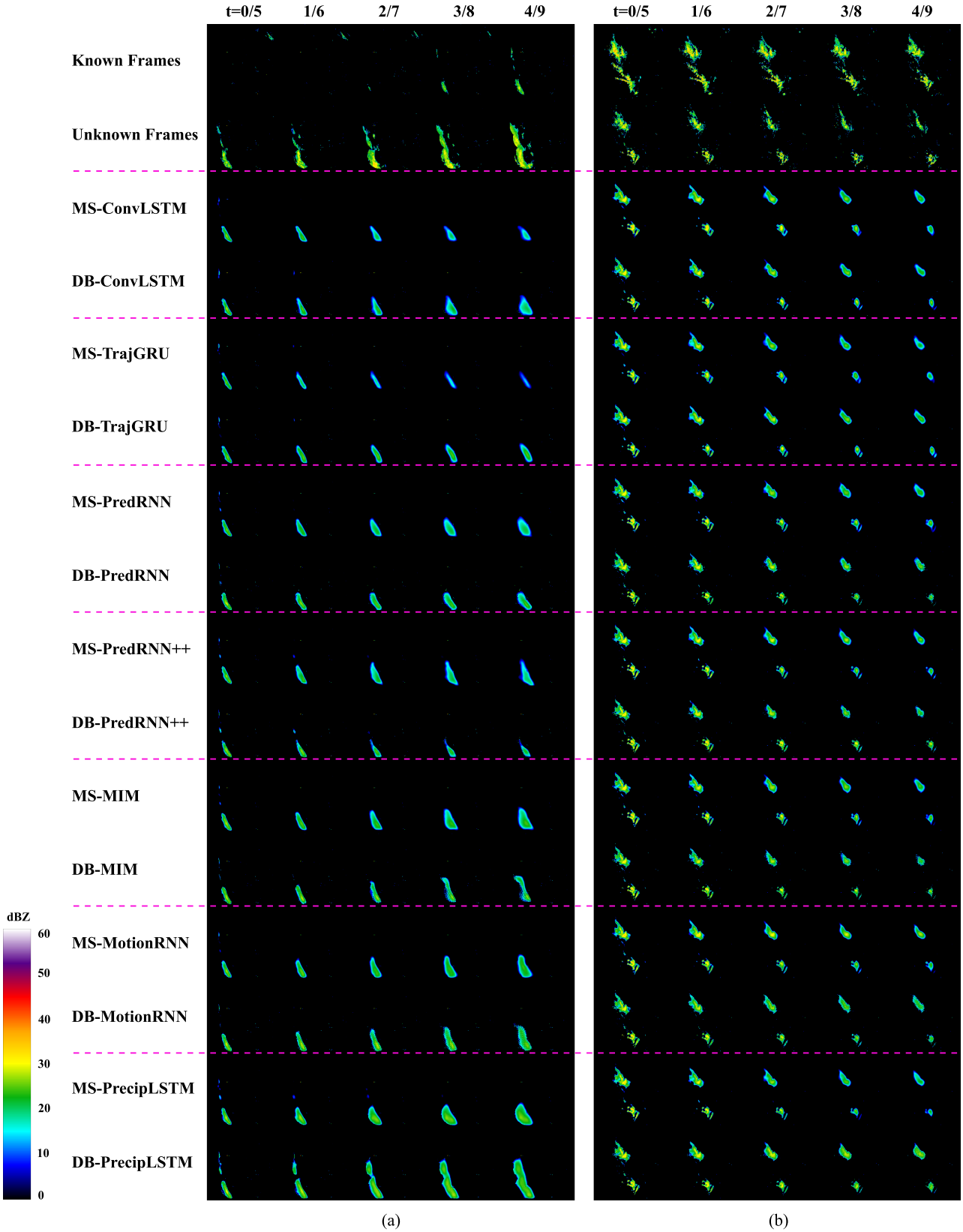

Fig. 6. Cases study of instantaneous precipitation starting at 2016-02-22 09:35:00 UTC+1 (left) and 2016-02-25 17:50:00 UTC+1 (right) of Germany. The first row is known historical frames, the second row is unknown future frames, and the other rows are predicted frames. (a) Process of increasing precipitation moving northward. (b) Process of decreasing precipitation moving eastward.

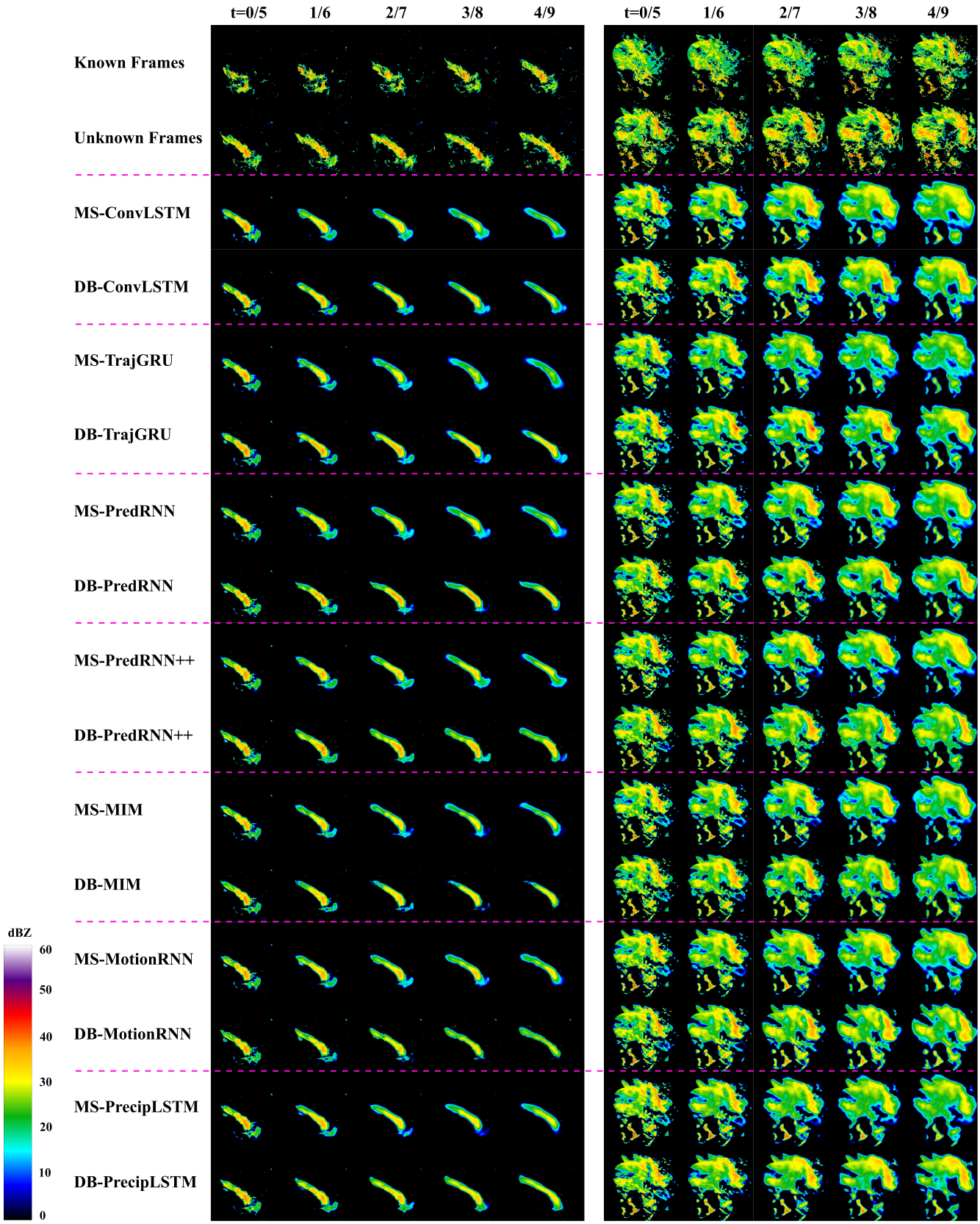

Fig. 7. Cases study of instantaneous precipitation starting at 2017-06-22 08:15:00 UTC+1 (left) and 2017-07-12 10:05:00 UTC+1 (right) of Germany. The first row is known historical frames, the second row is unknown future frames, and the other rows are predicted frames. (a) Process of the narrow precipitation band moving eastward. (b) Process of the broad precipitation cluster spreading eastward.

TABLE VI
COMPARISON WITH STAGED TRAINING MODELS ON HKO-7

Model	CSI-0.5 ↑	CSI-5 ↑	CSI-30 ↑	B-MSE ↓
MS-ConvLSTM [32]	0.668	0.519	0.201	268.8
MS-ConvLSTM-S	0.671	0.524	0.214	263.8
DB-ConvLSTM	**0.679**	**0.536**	**0.243**	**252.0**
MS-PredRNN [32]	0.677	0.534	0.236	247.4
MS-PredRNN-S	0.680	0.540	0.246	244.5
DB-PredRNN	**0.687**	**0.557**	**0.272**	**237.4**

MS-RNN-S means MS-RNN that uses the staged training strategy [see Fig. 3(a)]. The boldface indicates the optimal metric value in the current subtable, and each subtable indicates the comparison between MS-RNN and DB-RNN.

TABLE VII
COMPARISON WITH THE STATE-OF-THE-ART MODELS ON HKO-7

Models	Basics	Params ↓	CSI-0.5 ↑	CSI-5 ↑	CSI-30 ↑
Earthformer [27]	Atten	7.2M	0.549	0.364	0.035
SimVP [44]	CNN	38.9M	0.585	0.446	0.161
LPT-QPN [53]	Atten	4.6M	0.602	0.409	0.092
MS-LSTM [26]	RNN	1.1M	0.675	0.532	0.200
DB-ConvLSTM	RNN	**0.5M**	0.679	0.536	0.243
DB-PrecipLSTM	RNN	2.2M	**0.691**	**0.562**	**0.285**

The bold fonts indicate the best between DB-RNN and other SOTA models. In order to highlight our DB-RNN.

the introduction of the deblurring network does improve the performance of MS-RNN (reduction of blur), and the end-to-end training strategy is better than the staged training strategy. The skip connections we introduce between the forecasting network and the deblurring network and our unique training method that enables the two networks to cooperate are the fundamental reasons why DB-RNN is superior to MS-RNN-S. Furthermore, when experimenting with MS-RNN-S, its tedious and time-consuming training and testing give us a headache.

H. Comparison With State-of-The-Art Models

We compare DB-RNN with four recent state-of-the-art models (Table VII). They adopt different architectures, among which Earthformer [27] and LPT-QPN [53] are attention (abbreviated as Atten) models, SimVP [44] is a convolutional model, and MS-LSTM [26] is a recursive model. We enlist a model with the worst performance, DB-ConvLSTM, and a model with the strongest performance, DB-PrecipLSTM. In terms of parameters, DB-ConvLSTM has the least, SimVP has the most, and other models are in the middle. In terms of performance, DB-PrecipLSTM is the strongest, Earthformer is the weakest, and other models are in the middle. It is worth noting that DB-ConvLSTM uses the smallest parameters to obtain the second-best performance and DB-PrecipLSTM uses the third-smallest parameters to obtain the best performance. In short, DB-RNN mostly outperforms all historical state-of-the-art models in terms of parameters and performance.

I. Exploration of Long Term Forecasts

To explore the performance of MS-RNN and DB-RNN for long-term forecasting, we conduct experiments on HKO-7. We use PrecipLSTM as the base RNN model and use HSS-10 and POD-10 as indicators. To save training time and training

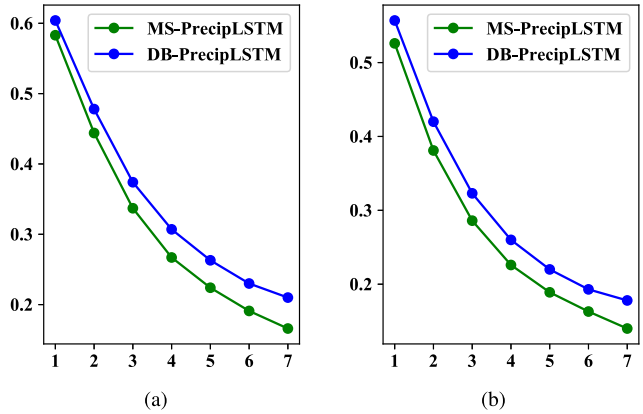

Fig. 8. Comparison of long-term prediction performance of MS-RNN and DB-RNN on HKO-7. The horizontal axis represents the forecast lead time in hours. (a) HSS-10. (b) POD-10.

TABLE VIII
ABLATION STUDY ON HKO-7

Model	L_{deb}	L_{gdl}	L_{adv}	CSI-10 ↑	B-MAE ↓
MS-PrecipLSTM [32]	✗	✗	✗	0.430	884.2
DB-PrecipLSTM	✓	✗	✗	0.439	874.6
DB-PrecipLSTM	✓	✗	✗	0.442	870.5
DB-PrecipLSTM	✓	✓	✗	0.443	868.1
DB-PrecipLSTM	✓	✓	✓	**0.448**	**862.3**

The boldface indicates the optimal DB-PrecipLSTM.

resources, we reuse the weights of models from previous experiments (see Section V-B). Specifically, we use fixed and trained models (MS-PrecipLSTM and DB-PrecipLSTM) to perform forecast tasks with different lead times, which is easily achieved through multiple recursions.

We have both MS-PrecipLSTM and DB-PrecipLSTM perform 1–7 h forecasts (see Fig. 8). The performance of both drops sharply in the initial stage and then slows down, which is somewhat similar to Fig. 4. However, the performance degradation of DB-PrecipLSTM is slower than that of MS-PrecipLSTM, which becomes more obvious as the lead time increases, which is consistent with the analysis of Fig. 4. Although the autoregressive structure of DB-RNN accumulates error (blur), deblurring is also accumulated. Nonetheless, the performance of DB-RNN is not optimistic in the long term, which may be caused by the use of only a single radar source and the lack of physical constraints. The introduction of multimodal data and physical constraints may alleviate this situation in the future.

J. Ablation Studies

To demonstrate the role of each component in our proposed DB-RNN, we sequentially add the deblurring loss (L_{deb}), GDL (L_{gdl}), and adversarial loss (L_{adv}) to MS-RNN (L_{for}). We use PrecipLSTM [30] as the basic model and perform experiments on the HKO-7 dataset. The results in Table VIII show that these components all play a positive role, among which the introduction of the deblurring loss has the largest improvement, the introduction of adversarial loss has the intermediate improvement, and the introduction of GDL has the smallest improvement. In addition, to verify the effectiveness of the dynamic weight

setting during the training process of DB-RNN, we fix λ_1 and λ_2 in (7) to 1 during training (row 2 of Table VIII). The results show that the dynamic loss weight strategy (row 3 of Table VIII) is slightly better than the static loss weight strategy. This suggests that it might be better to split the network into forecasting and deblurring parts and let them focus on their roles during training.

VI. Discussion

In the context of computer vision, precipitation nowcasting belongs to the subtask of weather forecasting, and weather forecasting belongs to the subtask of video prediction. They all belong to spatiotemporal prediction tasks, and most of the models in these domains are common. In addition, DB-RNN has opened up the connection with video deblurring. Therefore, we can infer that DB-RNN can also be extended to these domains. In this section, we mainly discuss applications and corresponding limitations of DB-RNN in different domains.

A. Weather Forecasting

1) This article only conducts experiments on two precipitation datasets with different climates: one with a subtropical monsoon climate and one with a temperate maritime climate. We believe that DB-RNN is capable of handling data from other climates. Different climate types bring different distributions of precipitation levels, but from the experiments of DB-RNN on HKO-7, we can see that DB-RNN performs well whether it is light rain, moderate rain, or heavy rain. Therefore, as long as the data are sufficient, DB-RNN can still perform well.
2) Although temperature, pressure, and wind speed do not change much in the short term [39], the blur problem in the forecast of these meteorological elements still exists, and DB-RNN can also be used to deal with these types of blur.
3) Recently, data-driven large meteorological models have emerged endlessly, such as FourCastNet [23], Swin-VRNN [78], Pangu-Weather [21], and ClimaX [79], etc. They learn underlying meteorological laws from massive historical climate data of Earth, which makes them excellent at modeling long-term climate dynamics, even beating traditional numerical models. However, the training data they use have low temporal and spatial resolutions, which makes it difficult to perform short-term high-resolution forecasting tasks, such as precipitation nowcasting. In short, we believe that small models, such as DB-RNN, are more suitable for short-term local weather forecasting, while data-driven large models are more suitable for long-term global weather forecasting. In operational weather forecasting scenarios, DB-RNN can be used as a supplement to numerical forecasting and large model forecasting systems.

B. Video Prediction and Video Deblurring

1) In addition to weather forecasting, there are also some other spatiotemporal prediction tasks, such as traffic flow prediction, robot movement prediction, human movement prediction, etc. Accurate predictions will benefit human life. To cope with future uncertainties, models in these domains also choose fuzzy predictions to minimize losses. That is, the blur problem is a recognized thorny problem in the video prediction domain. For models using RNN structures in the video prediction domain, the introduction of the deblurring network of DB-RNN is applicable. For models using non-RNN structures in the video prediction domain, the introduction of deblurring loss of DB-RNN is applicable. While these bring benefits, they introduce additional computational costs, and a tradeoff between performance and cost needs to be considered.
2) Similar to video prediction tasks, models in the video deblurring domain also have various architectures. For video deblurring models using RNN structures or RNN-like structures, the deblurring network design of DB-RNN is worth learning from. This may not apply to video deblurring models with non-RNN structures, but the adversarial and gradient regularization loss terms should play a corresponding deblurring role. We believe that DB-RNN can also perform well on deblurring datasets such as MC-Blur [80] and RWBI [81], which we leave to future work.

VII. Conclusion and Future Work

This article explores the introduction of the deblurring network and deblurring loss into the forecasting network and proposes the DB-RNN. Both the forecasting network and the deblurring network are composed of the advanced multiscale model MS-RNN. The deblurring loss has three parts, one of which comes from the deblurring network, and the other two parts consist of the GDL and adversarial loss. In addition, we introduce additional skip connections between the two MS-RNNs and design a progressive training strategy to make them work together. We perform quantitative and qualitative comparative experiments on two large-scale radar datasets called HKO-7 and DWD-12 respectively, and the results demonstrate that the predictions of DB-RNN are clearer, more accurate, and more focused on heavier rain than MS-RNN. In addition, we analyze the complexity and scalability of DB-RNN, compare it with historical state-of-the-art models and staged training models, and explore its application for long-term predictions. All the results demonstrate the superiority of DB-RNN. Next, ablation experiments demonstrate the effectiveness of the three-part deblurring design. Finally, we discuss the applications and limitations of DB-RNN in other domains.

Autoregression is a double-edged sword, which can lead to the accumulation of errors and the accumulation of deblurring. We flexibly employ the advantages of autoregression to cover up its disadvantages. Recently, some nonautoregressive models, such as SimVP and MIMO-VP, have begun to appear. They adopt an end-to-end approach instead of a recursive approach. Although this does not lead to error accumulation, it may cause training difficulties because the network design does not take time dependence into account. In conclusion, the tradeoff between autoregressive and nonautoregressive models requires further research.

Since the MSE (or MAE) loss always leads to the average of the future state, the GAN models choose to add adversarial regular terms to the loss function to combat ambiguity. In addition, some models also introduce regular terms such as GDL, structural similarity, and perceptual loss in the loss function. The regular term limits the size of the neural network solution space, making it easier for the model to obtain approximate solutions. DB-RNN also draws on this principle. Whether there are better reconstruction losses to replace the MSE (or MAE) and whether there are more powerful regularization terms are questions worth exploring in the future.

REFERENCES

[1] J. Thuemmel et al., "Inductive biases in deep learning models for weather prediction," 2023, *arXiv:2304.04664*.

[2] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[3] Z. Xu, Y. Wang, M. Long, and J. Wang, "PredCNN: Predictive learning with cascade convolutions," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 2940–2947.

[4] C. K. Sønderby et al., "MetNet: A neural weather model for precipitation forecasting," 2020, *arXiv:2003.12140*.

[5] R. Castro, Y. M. Souto, E. Ogasawara, F. Porto, and E. Bezerra, "STConvS2S: Spatiotemporal convolutional sequence to sequence network for weather forecasting," *Neurocomputing*, vol. 426, pp. 285–298, 2021.

[6] F. Chirigati, "Accurate short-term precipitation prediction," *Nature Comput. Sci.*, vol. 1, no. 11, pp. 709–709, 2021.

[7] C. J. Gamboa-Villafruela, J. C. Fernández-Alvarez, M. Márquez-Mijares, A. Pérez-Alarcón, and A. J. Batista-Leyva, "Convolutional LSTM architecture for precipitation nowcasting using satellite data," *Environ. Sci. Proc.*, vol. 8, no. 1, 2021, Art. no. 33.

[8] X. Shi et al., "Deep learning for precipitation nowcasting: A benchmark and a new model," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5622–5632.

[9] 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.

[10] M. R. Ehsani, A. Zarei, H. V. Gupta, K. Barnard, E. Lyons, and A. Behrangi, "NowCasting-Nets: Representation learning to mitigate latency gap of satellite precipitation products using convolutional and recurrent neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4706021.

[11] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, and N. Carvalhais, "Deep learning and process understanding for data-driven earth system science," *Nature*, vol. 566, no. 7743, pp. 195–204, 2019.

[12] Q. Yan, F. Ji, K. Miao, Q. Wu, Y. Xia, and T. Li, "Convolutional residual-attention: A deep learning approach for precipitation nowcasting," *Adv. Meteorol.*, vol. 2020, pp. 1–12, 2020.

[13] Z. Ma, H. Zhang, and J. Liu, "Focal frame loss: A simple but effective loss for precipitation nowcasting," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 6781–6788, 2022.

[14] C. Luo, X. Li, and Y. Ye, "PFST-LSTM: A spatiotemporal LSTM model with pseudoflow prediction for precipitation nowcasting," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 843–857, 2020.

[15] L. Sevilla-Lara, Y. Liao, F. Güney, V. Jampani, A. Geiger, and M. J. Black, "On the integration of optical flow and action recognition," in *Proc. German Conf. Pattern Recognit.*, 2018, pp. 281–297.

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

[17] 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.

[18] L. Han, Y. Zhao, H. Chen, and V. Chandrasekar, "Advancing radar nowcasting through deep transfer learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 4100609.

[19] S. Ravuri et al., "Skilful precipitation nowcasting using deep generative models of radar," *Nature*, vol. 597, no. 7878, pp. 672–677, 2021.

[20] F. Zhang, X. Wang, and J. Guan, "A novel multi-input multi-output recurrent neural network based on multimodal fusion and spatiotemporal prediction for 0–4 hour precipitation nowcasting," *Atmosphere*, vol. 12, no. 12, 2021, Art. no. 1596.

[21] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, "Accurate medium-range global weather forecasting with 3D neural networks," *Nature*, vol. 619, no. 7970, pp. 533–538, 2023.

[22] L. Espeholt et al., "Deep learning for twelve hour precipitation forecasts," *Nature Commun.*, vol. 13, no. 1, pp. 1–10, 2022.

[23] J. Pathak et al., "FourcastNet: A global data-driven high-resolution weather model using adaptive fourier neural operators," 2022, *arXiv:2202.11214*.

[24] R. Lam et al., "Learning skillful medium-range global weather forecasting," *Science*, vol. 382, no. 6677, pp. 1416–1421, 2023. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.adi2336

[25] K. Chen et al., "FengWu: Pushing the skillful global medium-range weather forecast beyond 10 days lead," 2023, *arXiv:2304.02948*.

[26] Z. Ma, H. Zhang, and J. Liu, "MS-LSTM: Exploring spatiotemporal multiscale representations in video prediction domain," *Appl. Soft Comput.*, vol. 147, 2023, Art. no. 110731.

[27] Z. Gao et al., "Earthformer: Exploring space-time transformers for earth system forecasting," in *Proc. Conf. Neural Inf. Process. Syst.*, 2022, pp. 25390–25403.

[28] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, "PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017, pp. 879–888.

[29] Y. Wang, J. Zhang, H. Zhu, M. Long, J. Wang, and P. S. Yu, "Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9154–9162.

[30] Z. Ma, H. Zhang, and J. Liu, "PrecipLSTM: A meteorological spatiotemporal LSTM for precipitation nowcasting," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4109108.

[31] H. Wu, Z. Yao, J. Wang, and M. Long, "MotionRNN: A flexible model for video prediction with spacetime-varying motions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15435–15444.

[32] Z. Ma, H. Zhang, and J. Liu, "MS-RNN: A flexible multi-scale framework for spatiotemporal predictive learning," 2022, *arXiv:2206.03010*.

[33] S. Oprea et al., "A review on deep learning techniques for video prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 2806–2826, Jun. 2022.

[34] K. Zhang et al., "Deep image deblurring: A survey," *Int. J. Comput. Vis.*, vol. 130, no. 9, pp. 2103–2130, 2022.

[35] H. Liu et al., "Video super-resolution based on deep learning: A comprehensive survey," *Artif. Intell. Rev.*, vol. 55, no. 8, pp. 5981–6035, 2022.

[36] Z. Wang, J. Chen, and S. C. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3365–3387, Oct. 2021.

[37] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *Proc. Int. Conf. Learn. Represent.*, 2016.

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

[39] Z. Ma, H. Zhang, and J. Liu, "MM-RNN: A multimodal RNN for precipitation nowcasting," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4101914.

[40] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2015, pp. 234–241.

[41] G. Ayzel, T. Scheffer, and M. Heistermann, "RainNet v1.0: A convolutional neural network for radar-based precipitation nowcasting," *Geosci. Model Develop.*, vol. 13, no. 6, pp. 2631–2644, 2020.

[42] J. G. Fernández and S. Mehrkanoon, "Broad-UNet: Multi-scale feature learning for nowcasting tasks," *Neural Netw.*, vol. 144, pp. 419–427, 2021.

[43] L. Han, H. Liang, H. Chen, W. Zhang, and Y. Ge, "Convective precipitation nowcasting using U-Net model," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 4103508.

[44] Z. Gao, C. Tan, L. Wu, and S. Z. Li, "SimVP: Simpler yet better video prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3170–3180.

[45] Y. Wang, Z. Gao, M. Long, J. Wang, and S. Y. Philip, "PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5123–5132.

[46] C. Zhang, X. Yang, Y. Tang, and W. Zhang, "Learning to generate radar image sequences using two-stage generative adversarial networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 3, pp. 401–405, Mar. 2020.

[47] Y. Kim and S. Hong, "Very short-term rainfall prediction using ground radar observations and conditional generative adversarial networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 4104308.
[48] C. Luo, X. Li, Y. Ye, S. Feng, and M. K. Ng, "Experimental study on generative adversarial network for precipitation nowcasting," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5114220.
[49] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[50] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2020 .
[51] C. Bai, F. Sun, J. Zhang, Y. Song, and S. Chen, "Rainformer: Features extraction balanced network for radar-based precipitation nowcasting," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 4023305.
[52] S. Ning et al., "MIMO is all you need: A strong multi-in-multi-out baseline for video prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 2, 2023, pp. 1975–1983.
[53] D. Li et al., "LPT-QPN: A lightweight physics-informed transformer for quantitative precipitation nowcasting," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4107119.
[54] A. Lucas, S. Lopez-Tapia, R. Molina, and A. K. Katsaggelos, "Generative adversarial networks and perceptual losses for video super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3312–3327, Jul. 2019.
[55] W. Li, X. Tao, T. Guo, L. Qi, J. Lu, and J. Jia, "MuCAN: Multi-correspondence aggregation network for video super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 335–351.
[56] Y. Jo, S. W. Oh, J. Kang, and S. J. Kim, "Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3224–3232.
[57] Z. He, D. He, X. Li, and J. Xu, "Unsupervised video satellite super-resolution by using only a single video," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2020, Art. no. 6000905.
[58] X. Ying, L. Wang, Y. Wang, W. Sheng, W. An, and Y. Guo, "Deformable 3D convolution for video super-resolution," *IEEE Signal Process. Lett.*, vol. 27, pp. 1500–1504, 2020.
[59] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-revealing deep video super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4472–4480.
[60] J.-Y. Lin, Y.-C. Chang, and W. H. Hsu, "Efficient and phase-aware video super-resolution for cardiac MRI," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2020, pp. 66–76.
[61] W. Sun, J. Sun, Y. Zhu, and Y. Zhang, "Video super-resolution via dense non-local spatial-temporal convolutional network," *Neurocomputing*, vol. 403, pp. 1–12, 2020.
[62] D. Li, Y. Liu, and Z. Wang, "Video super-resolution using non-simultaneous fully recurrent convolutional network," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1342–1355, Mar. 2019.
[63] T. Isobe, X. Jia, S. Gu, S. Li, S. Wang, and Q. Tian, "Video super-resolution with recurrent structure-detail network," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 645–660.
[64] C. Zhu et al., "Deep recurrent neural network with multi-scale bi-directional propagation for video deblurring," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 3598–3607.
[65] 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. 4681–4690.
[66] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8878–8887.
[67] M. Cao, Y. Fan, Y. Zhang, J. Wang, and Y. Yang, "VDTR: Video deblurring with transformer," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 1, pp. 160–171, Jan. 2023.
[68] J. Liang et al., "Recurrent video restoration transformer with guided deformable attention," in *Proc. Conf. Neural Inf. Process. Syst.*, 2022, pp. 378–393.
[69] H. Zhang, H. Xie, and H. Yao, "Spatio-temporal deformable attention network for video deblurring," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 581–596.
[70] J. Guo and H. Chao, "Building an end-to-end spatial-temporal convolutional network for video super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4053–4060.
[71] X. Zhu, Z. Li, X.-Y. Zhang, C. Li, Y. Liu, and Z. Xue, "Residual invertible spatio-temporal network for video super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 5981–5988.
[72] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. Cambridge, MA, USA: MIT Press, 1994.
[73] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.
[74] Z. Lin, M. Li, Z. Zheng, Y. Cheng, and C. Yuan, "Self-attention convLSTM for spatiotemporal prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11531–11538.
[75] Z. Chai, Z. Xu, and C. Yuan, "MoDeRNN: Towards fine-grained motion details for spatiotemporal predictive learning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 4658–4662.
[76] Z. Chai, Z. Xu, Y. Bail, Z. Lin, and C. Yuan, "CMS-LSTM: Context embedding and multi-scale spatiotemporal expression LSTM for predictive learning," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2022, pp. 01–06.
[77] Y. Wang et al., "PredRNN: A recurrent neural network for spatiotemporal predictive learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2208–2225, Feb. 2023.
[78] Y. Hu, L. Chen, Z. Wang, and H. Li, "SwinvRNN: A data-driven ensemble forecasting model via learned distribution perturbation," *J. Adv. Model. Earth Syst.*, vol. 15, no. 2, 2023, Art. no. e2022MS003211.
[79] T. Nguyen, J. Brandstetter, A. Kapoor, J. K. Gupta, and A. Grover, "ClimaX: A foundation model for weather and climate," in *Proc. Int. Conf. Learn. Representations*, 2023.
[80] K. Zhang et al., "MC-Blur: A comprehensive benchmark for image deblurring," *IEEE Trans. Circuits Syst. Video Technol.*, early access, doi: 10.1109/TCSVT.2023.3319330.
[81] K. Zhang et al., "Deblurring by realistic blurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2737–2746.

Zhifeng Ma received the master's degree in applied statistics from Lanzhou University, Lanzhou, China, in 2020. He is currently working toward the Ph.D. degree in computer science and technology with the Harbin Institute of Technology, Harbin, China.

His research interests include precipitation nowcasting and video prediction.

Hao Zhang received the Ph.D. degree in information security from the University of Science and Technology of China, Hefei, China, in 2014.

He is currently an Associate Researcher with the Harbin Institute of Technology, Harbin, China. His research interests include deep learning applications and federated learning.

Jie Liu (Fellow, IEEE) received the Ph.D. degree in electrical engineering and computer science from the University of California at Berkeley, Berkeley, CA, USA, in 2001.

He is currently a Chair Professor with the Harbin Institute of Technology, Shenzhen, China. His research interests include Internet of Things and ambient intelligence.