Accurate and Clear Quantitative Precipitation Nowcasting Based on a Deep Learning Model with Consecutive Attention and Rain-Map Discrimination

ASHESH, A CHIA-TUNG CHANG, B BUO-FU CHEN, B HSUAN-TIEN LIN, A BOYO CHEN, B AND TRENG-SHI HUANG C

A Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan
B Center for Weather Climate and Disaster Research, National Taiwan University, Taipei, Taiwan
C Central Weather Bureau, Taipei, Taiwan

(Manuscript received 1 November 2021, in final form 28 April 2022)

ABSTRACT: Deep learning models are developed for high-resolution quantitative precipitation nowcasting (QPN) in Taiwan up to 3 h ahead. Many recent works aim to accurately predict relatively rare high-rainfall events with the help of deep learning. This rarity is often addressed by formulations that reweight the rare events. However, these formulations often carry a side effect of producing blurry rain-map nowcasts that overpredict in low-rainfall regions. Such nowcasts are visually less trustworthy and practically less useful for forecasters. We fix the trust issue by introducing a discriminator that encourages the model to generate realistic rain maps without sacrificing the predictive accuracy of rainfall extremes. Moreover, with consecutive attention across different hours, we extend the nowcasting time frame from typically 1 to 3 h to further address the needs for socioeconomic weather-dependent decision-making. By combining the discriminator and the attention techniques, the proposed model based on the convolutional recurrent neural network is trained with a dataset containing radar reflectivity and rain rates at a granularity of 10 min and predicts the hourly accumulated rainfall in the next three hours. Model performance is evaluated from both statistical and case-study perspectives. Statistical verification shows that the new model outperforms the current operational QPN techniques. Case studies further show that the model can capture the motion of rainbands in a frontal case and also provide an effective warning of urban-area torrential rainfall in an afternoon-thunderstorm case, implying that deep learning has great potential and is useful in 0–3-h nowcasting.

KEYWORDS: Radars/radar observations; Nowcasting; Data science; Deep learning; Machine learning; Neural networks

1. Introduction

Short-term (usually referred to as <12 h) precipitation forecasting is one of the most important weather forecasting topics. Large-scale, fine-grained, and real-time forecasts for the short term provide society with crucial information, such as road conditions, aviation weather reports, and flood alerts. The information, in turn, facilitates safer and better daily lives.

There are two extremes of short-term forecasts. For 6–12-h forecasts, numerical weather prediction (NWP) models driven by physics simulation generally provide more stable predictions due to proper use of the domain knowledge (Kain et al. 2010; Sun et al. 2014) and are usually considered superior to data-driven techniques. On the other hand, for 0–1-h forecasts, that is, quantitative precipitation nowcasting (QPN), data-driven techniques such as the radar echo extrapolation models are powerful solutions (Dixon and Wiener 1993; Germann and Zawadzki 2002, 2004; Chung and Yao 2020; Chang et al. 2021). These extrapolation-based models leverage the high temporal and spatial resolutions of radar maps whenever they are available and convert the radar reflectivity to the predicted rain rates for the QPN problem. Nevertheless, extrapolation-based models suffer from the difficulties of accurately converting to the rain rates and capturing the growth and decay of storms, such as anticipating storm motions at a larger lead time (i.e., at the second and third hours). A possible approach for improving longer lead time predictive ability is by combining the radar-based nowcast with the NWP model (Chung and Yao 2020; Radhakrishnan and Chandrasekar 2020). For instance, Radhakrishnan and Chandrasekar (2020) used both a high-resolution dual-polar radar storm tracking technique and a mesoscale NWP model with 3DVAR radar assimilation to predict severe storms in Texas.

The deep learning community has recently shown great interest in the QPN problem (Shi et al. 2015, 2017; Tran and Song 2019; Franch et al. 2020; Ayzel et al. 2020; Sønderby et al. 2020; Trebing et al. 2021; Espeholt et al. 2021; Ravuri et al. 2021). Shi et al. (2015) took the radar reflectivity data and modeled the QPN problem as a spatiotemporal sequence prediction problem on the radar reflectivity sequence. They modeled the sequence prediction problem by a deep model based on the convolutional long-term short-term memory (ConvLSTM) architecture. A follow-up work (Shi et al. 2017) replaced ConvLSTM with a novel architecture, trajectory gated recurrent units (TrajGRU), which was claimed to capture the location-variant features better. The TrajGRU model outperformed the extrapolation-based models, making it one of the state-of-the-art deep learning QPN solutions based only on radar reflectivities. The promising performance of these deep learning models reveals the potential of applying deep learning to the QPN problem. Both works (Shi et al. 2015, 2017) introduced a weighted loss that gives more weight to the locations with higher radar reflectivity to achieve better predictions for the rare heavy-rainfall events and used a mask to ignore locations whose radar reflectivity is below a threshold.

Those models make predictions on the radar reflectivity Z and then convert the predictions to the rain R maps using...
the \( R(Z) \) relationship formula. However, it leads to uncertainty of rainfall estimates because the \( R(Z) \) relationship varies with time, location, and other factors, such as echo-top height and relative humidity (Alfieri et al. 2010; Wu et al. 2018; Chang et al. 2021). Thus, relying on the coarse \( R(Z) \) formula on top of using only the radar reflectivity data seriously restricts the practical applicability of the TrajGRU model (Shi et al. 2017). In other words, it remains unanswered whether the TrajGRU model is the most competitive solution for more complete datasets that contain rainfall data.

Besides the issue of using only the radar data, two other critical demands are yet to be addressed. First, meteorologists often demand the models to predict high rain rates accurately for disaster prevention while simultaneously predicting reasonable rain rates in drizzle (low rainfall) regions. Current deep learning models achieve decent performance on high-rainfall regions by reweighting in the objective function but output a larger raining area as a side effect. Specifically, this kind of prediction contains many locations with small but nonzero predicted rain rates, which give a hazy appearance on the predicted rain map. This is a natural consequence of giving low weights to low-rainfall regions and using a mask to ignore low-rainfall locations. The hazziness in prediction is not desirable. In particular, the hazziness makes meteorologists skeptical about using the deep learning model since the conventional QPN models in operation do not suffer from such artifacts (Lakshmanan et al. 2003; Lakshmanan and Smith 2010; Chung and Yao 2020).

Tran and Song (2019) tried to improve the ConvLSTM and TrajGRU models by coupling the objective function with another loss term, such as (the negative) structural similarity (SSIM) and multiscale SSIM, to pull the generated images closer to the actual images in terms of image quality. Franch et al. (2020) proposed a solution based on model stacking with another convolutional neural network to improve the prediction accuracy on different rain regimes. Both studies reduced some haziness, but usually, their generated images were still easy for a human (not even a meteorologist) to distinguish between a realistic rain map and the predicted (generated) rain map, where visual blurriness appears to be the primary distinguishing feature.

In addition, most existing deep learning models (Shi et al. 2017; Tran and Song 2019; Franch et al. 2020; Ayzel et al. 2020; Trebing et al. 2021) provided only a first-hour prediction, while meteorologists demand accurate predictions with a longer period. Extending the prediction period from one hour to three hours can be helpful in many other hydrometeorological applications, such as flood prediction and river stage forecasting. The extension is also critical toward increasing the reaction time for disaster prevention and management. In comparison with the conventional extrapolation-based models, it appears that deep learning models can potentially leverage their stronger predictive power to achieve longer prediction periods, but the potential has not been fully explored.

This work makes the QPN problem more realistic by compiling a complete and ready-to-use dataset containing both the column maximum radar reflectivity and the rain rates in Taiwan. Moreover, a novel deep learning model is designed to address both demands: (i) to produce predictions that meteorologists can trust in both low- and high-rainfall regions and (ii) to improve the prediction quality for longer periods (i.e., 1–3 h). Thus, this study touches upon two important techniques from deep learning, namely adversarial learning and attention. Adversarial learning has been greatly popularized by generative adversarial networks (GANs; Goodfellow et al. 2014), which are primarily used for image generation to guide generative models toward producing realistic images. The discriminator design within GANs has been successfully used in other applications, such as enhancing the clarity of generated images (Kwon and Park 2019; Vondrick et al. 2016). Attention-based approaches have found decent popularity in the computer vision community (Woo et al. 2018; Li et al. 2020). Their ability to focus selectively on an image is helpful in cases where features of interest span a small number of pixels and their occurrences are rare (Lim et al. 2019; Bai et al. 2020; Zhang et al. 2020).

The proposed deep learning model is applied to radar data and rain rates for the extended QPN problem of up to three hours. A discriminator module, which distinguishes between the realistic and generated rain maps, nudges the deep model to generate more realistic rain maps. We observe both qualitatively and quantitatively that the discriminator indeed leads to less blurry rain maps without compromising the predictive performance of the deep model. To further produce accurate 0–3-h predictions, an attention module is integrated to create focus regions on the rain maps that can be carried from the earlier hours to the later ones, and the focus regions are used to rescale the predicted rainfall.

Parallel to our work, there are two other notable deep learning-based approaches for rainfall prediction (Espeholt et al. 2021; Ravuri et al. 2021). Espeholt et al. (2021) proposed large context neural networks that integrate radar observation, satellite observation, and even initial fields \( (U, V, T, Q, \text{ and } P) \) of NWP models to generate large-scale (synoptic scale) precipitation forecasting 12 h into the future. As compared with their larger domain over a 7000 km \( \times \) 2500 km region of the continental United States and longer forecast duration, the current study primarily uses radar observations and tackles the conventional QPN topic that focuses on mesoscale to storm-scale extreme rainfalls. Ravuri et al. (2021) developed deep generative models predicting 0–90-min rainfall in the United Kingdom. Our work is similar to theirs because a discriminator module is proposed to obtain less blurry and realistic rain maps. However, this study further employs the attention module to extend the prediction duration to three hours.

This paper is organized as follows. Sections 2 and 3 introduce the data and problem setup and describe the architecture of the deep learning QPN model. Section 4 evaluates the performance of the proposed model based on statistical verification. Two case studies are carried out in section 5 to discuss how useful the model is in real cases of extreme rainfall. The conclusive remarks and future directions are in section 6.
2. Data and problem setup

a. Input data and the prediction target

To make research on the QPN problem more realistic, we compile a dataset consisting of two collections—rain rates and column maximum radar reflectivity in Taiwan (Fig. 1). Both datasets are produced by the in-operation Quantitative Precipitation Estimation and Segregation Using Multiple Sensor (QPESUMS) system (Chang et al. 2021) from the Central Weather Bureau (CWB) of Taiwan. To the best of our knowledge, this study is the first deep learning work using both time-synchronized radar and rainfall data to tackle the 0–3-h QPN. The QPESUMS system integrates observations from multiple mixed-band weather radars (Fig. 1a) and rain gauges to produce high-resolution (≈1 km) and rapid-update (10 min) rainfall and severe storm monitoring and prediction products. Chang et al. (2021) documented a significant effort with data quality control to understand the precipitation characteristics and then developed a series of QPE methodologies that could be refined over time for each radar system. Namely, rain rates derived from individual radars via different methodologies depending on the radar types are more realistic than the values estimated by the simple $R(Z)$ formula (Chang et al. 2021).

At each 10-min time, the QPESUMS rain rates (mm h$^{-1}$) is a 2D map of shape 561 × 441, with a horizontal resolution of 0.0125° (Fig. 1c). On the other hand, each 3D scan of the radar reflectivity data (measured with dBZ) possesses images of size 561 × 441 with 21 levels in height. We calculated the column maximum radar reflectivity (Fig. 1b) in this study by converting the 3D volume to a 2D map by taking the maximum value of the radar reflectivity over the 21 levels.

The training–validation–test division of the data is described as follows. Both the rain rates and radar reflectivity have a time resolution of 10 min and comprise approximately 203,000 frames ranging from 1 January 2015 to 31 December 2018. We divide the data into training, test, and validation subsets by their time stamp. All data from the beginning of 2015 to the end of 2017 belong to the training set. The first 15 days of each month in 2018 belong to the testing set, and the last 15 days of each month in 2018 belong to the validation set. Note that the predictability in radar/rain signals (i.e., convective storms) vanishes rapidly with time. The fact that most deep learning models developed until now predict for a maximum of 12 h into the future is a testament to this fact. Thus, although data of the first 15 days and the last 15 days of a month may share the same seasonal climatology, there is no other direct relationship between the two, making them suitable candidates to be taken as the validation and testing sets covering all seasons.

The data preprocessing of the prediction target $Y_t$ is described. Let $y_t$ denotes the 2D rain-rate map for the time $[t − 10 \text{ min}, t)$, where the map is of size $J \times K$; subsequently, the prediction target is defined as

$$Y_t = \frac{1}{6} \sum_{i=0}^{5} y_{t-i}$$

storing the average hourly rain rates. Namely, $Y_t$ represents the accumulated hourly rainfall within $[t − 60 \text{ min}, t)$, also a $J \times K$ map. This study decides to predict the hourly rainfall in the future 0–3 h ahead instead of the 10-min rates as the input $y_t$, because hourly rainfall product is practical and widely used by government agencies and emergency managers in Taiwan for flood and mudslide warnings and water resource management. Some studies (e.g., Espeholt et al. 2021) handling large input data containing sufficient synoptic weather information may yield predictions of longer lead times up to 12 h. However, another reason for selecting the prediction window of 0–3 h in this work is that the intrinsic predictability of mesoscale to storm-scale extreme rainfall vanishes rapidly with time (typically <6 h).
In addition, the model developed in this study is evaluated against the benchmark QPESUMS extrapolation technique, which provides the first-hour rainfall prediction made by a radar extrapolation based on upgraded optical-flow-based models (Lakshmanan et al. 2003; Lakshmanan and Smith 2010). The extrapolation technique first analyzes the moving vector of each convection cell (radar echo) from the 2D radar map and then adds the vectors to the previous rain map. Because the technique does not consider the evolution of convective cells, it cannot reflect the growth and decline of weather systems. Thus, the effective forecast time of the benchmark model is only up to 1 h. This benchmark model is currently used in operation by forecasters in CWB.

b. Problem setup and verification methods

Let $x_t$, $y_t$, and $Y_t$ denote 2D radar-reflectivity map, 2D rainfall map, and target hourly rainfall at time $t$, respectively. The extended QPN problem that will be studied aims to predict $Y_t$ for the next three hours, given the radar and rain maps ($x_t$ and $y_t$) in the past hour. That is, the model needs to take the maps in the past hour, namely $[Y_{t-50min}, Y_{t-40min}, \ldots, Y_{t-0min}]$, and $[x_{t-50min}, x_{t-40min}, \ldots, x_{t-0min}]$ as the input and accurately predict the rainfall in the next three hours: $Y_{t+3h} = [\hat{Y}_{t+1h}, \hat{Y}_{t+2h}, \hat{Y}_{t+3h}]$. We denote the predictions made by the models as $\hat{Y}_{t+3h} = [\hat{Y}_{t+1h}, \hat{Y}_{t+2h}, \hat{Y}_{t+3h}]$ and we hope that $Y_{t+3h} \approx \hat{Y}_{t+3h}$.

One important measure for estimating the (dis)similarity between $Y_t$ and $\hat{Y}_t$ is called the weighted mean absolute error (WMAE). The measure $\text{WMAE}(Y_t, \hat{Y}_t)$ can be calculated by

$$\text{WMAE}(Y_t, \hat{Y}_t) = \frac{1}{J \times K} \sum_{j,k} W(Y_t[j,k])(|Y_t[j,k] - \hat{Y}_t[j,k]|),$$

where $j$, $k$ iterates over all $J$, $K$, respectively. The measure takes a given weight function, parameterized by $Y_t$, to emphasize the importance in heavy-rainfall regions. We adopt a domain-driven weight function that is similar to the one used in Shi et al. (2015, 2017):

$$W(r) = \begin{cases} 0 & r < 0.5 \\ 1 & 0.5 \leq r < 2 \\ 2 & 2 \leq r < 5 \\ 5 & 5 \leq r < 10 \\ 10 & 10 \leq r < 30 \\ 30 & 30 \leq r \end{cases}$$

Two other important measures are commonly used to assess how the model performs under different levels of rainfall: critical success index (CSI; Jolliffe and Stephenson 2012) and Heidke skill score (HSS; Hogan et al. 2010). The measures can be computed by first binarizing both $Y_t$ and $\hat{Y}_t$ with respect to a given threshold $\varphi$ to form binary labels and predictions. We can then summarize the labels and predictions as four numbers: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The CSI($Y_t, \hat{Y}_t$) is given by

$$\text{CSI}(Y_t, \hat{Y}_t) = \frac{TP}{TP + FN + FP}.$$  

The HSS is a metric commonly used in meteorology that accounts for class imbalance. It is defined as how much better the model prediction accuracy is than the standard forecast:

$$\text{HSS}(Y_t, \hat{Y}_t) = \frac{ACC - SF}{1 - SF},$$

$$\text{ACC} = \frac{TP + TN}{N},$$

$$\text{SF} = \frac{TP + FN}{N} \times \frac{TP + FP}{N} + \frac{TN + FN}{N} \times \frac{TN + FP}{N},$$

where $N$ is the number of total instances, ACC is the accuracy, SF is the standard forecast, defined as correct by chance with class proportion.

In addition, this study uses performance diagrams (Roebber 2009) to evaluate the QPN models (shown later in Figs. 8 and 12). The performance diagram exploits the geometric relationship between four measures of dichotomous forecast performance: the probability of detection ($POD = TP/(TP + FN)$; $y$ axis), success ratio ($SR = 1 - (FP/(TP + FP))$; $x$ axis), bias (the slope in the diagram), and CSI. For a perfect forecast, POD, SR, bias, and CSI approach unity and lie in the upper-right corner of the diagram. Deviations in a particular direction will indicate relative differences in POD, SR, and consequently, bias, and CSI.

3. Design of the deep learning QPN model

This section first gives a brief overview of the building blocks of our architecture: RNNs and CNNs, and next describes the three modules that make up our model (Figs. 2–4). We then describe the loss functions used in this work. Last, we give exact configurations of the different modules and other details relevant for training the model.

a. A brief introduction of ConvGRU

Recurrent neural networks (RNNs) were developed to handle the sequential data. They are designed to retain “memory” about previously seen data points in a data sequence. Gated recurrent unit (GRU; Cho et al. 2014) and long-term short-term memory (LSTM; Hochreiter and Schmidhuber 1997) are among the top RNN modules used today. On the other hand, convolutional neural networks (CNNs) were developed to extract information from 2D data types like images. Convolution operation, their essential feature, allows translation invariant feature generation and requires far fewer parameters than what is required by a fully connected layer. The output of a convolution filter is usually added by a separately trained bias vector and plugged into an activation function. The feature maps associated with every filter are concatenated together, constructing the input to the next layer.
For handling sequential imagelike data (e.g., the 2D rain maps), RNN and CNN were combined to yield ConvGRU, TrajGRU, and other similar modules (Shi et al. 2017). As we work with sequential 2D rain and radar maps in this study, ConvGRU is thus a natural choice to use in this setting. When dealing with a forecasting problem, an encoder–forecaster structure is often used, in which ConvGRU or similar modules are internally used. The encoder part gleans useful information from the past data and then passes it to the forecaster in the form of the initial state of the forecaster’s RNN modules. Then, the forecaster predicts the target sequence using just the information provided by the encoder. We describe it in more detail in section 3b.

b. Encoder–forecaster module

Following both seminal studies on the deep learning OCPN problem (Shi et al. 2015, 2017), we start our design with the same encoder–forecaster architecture. Our preliminary tests (not shown) examined the performance of the TrajGRU (Shi et al. 2017) on the current dataset but found that the trajectory part does not bring any additional benefits. We thus decided to use the simpler ConvGRU as our core RNN architecture.

As shown in Fig. 2, at time $t$, our ConvGRU encoder–forecaster module takes the rain rates of the last hour ($y_{t-50}$ to $y_t$) and column-maximum radar reflectivity ($x_{t-50}$ to $x_t$) as the input and generate a first-guess prediction of the hourly rainfall at time $t$, $t+1$ h, and $t+2$ h. The encoder part comprises three layers of GRU-based RNNs with one 2D convolution layer between every consecutive RNN pair (Fig. 2, left part). The convolution layer also downsamples the spatial dimension, allowing feature extraction at multiple scales. Furthermore, we used leaky rectified linear unit (LeakyReLU) (with negative_slope = 0.2) and sigmoid in both the encoder and forecaster. They have been applied inside the recurrent units in the same way as is the general practice. The forecaster has a similar structure with three RNNs with two 2D transposed convolution layers sandwiched between them (Fig. 2, right part). The transposed convolution layers perform upsampling in the forecaster part of the module. The detailed neural network configurations of the encoder–forecaster module can be found in Table 1.

For handling sequential imagelike data (e.g., the 2D rain maps), RNN and CNN were combined to yield ConvGRU, TrajGRU, and other similar modules (Shi et al. 2017). As we work with sequential 2D rain and radar maps in this study, ConvGRU is thus a natural choice to use in this setting. When dealing with a forecasting problem, an encoder–forecaster structure is often used, in which ConvGRU or similar modules are internally used. The encoder part gleans useful information from the past data and then passes it to the forecaster in the form of the initial state of the forecaster’s RNN modules. Then, the forecaster predicts the target sequence using just the information provided by the encoder. We describe it in more detail in section 3b.

b. Encoder–forecaster module

Following both seminal studies on the deep learning OCPN problem (Shi et al. 2015, 2017), we start our design with the same encoder–forecaster architecture. Our preliminary tests (not shown) examined the performance of the TrajGRU (Shi et al. 2017) on the current dataset but found that the trajectory part does not bring any additional benefits. We thus decided to use the simpler ConvGRU as our core RNN architecture.

As shown in Fig. 2, at time $t$, our ConvGRU encoder–forecaster module takes the rain rates of the last hour ($y_{t-50}$ to $y_t$) and column-maximum radar reflectivity ($x_{t-50}$ to $x_t$) as the input and generate a first-guess prediction of the hourly rainfall at time $t$, $t+1$ h, and $t+2$ h. The encoder part comprises three layers of GRU-based RNNs with one 2D convolution layer between every consecutive RNN pair (Fig. 2, left part). The convolution layer also downsamples the spatial dimension, allowing feature extraction at multiple scales. Furthermore, we used leaky rectified linear unit (LeakyReLU) (with negative_slope = 0.2) and sigmoid in both the encoder and forecaster. They have been applied inside the recurrent units in the same way as is the general practice. The forecaster has a similar structure with three RNNs with two 2D transposed convolution layers sandwiched between them (Fig. 2, right part). The transposed convolution layers perform upsampling in the forecaster part of the module. The detailed neural network configurations of the encoder–forecaster module can be found in Table 1.
c. Attention module

As shown in Fig. 3, the attention module produces three 2D attention maps $A_{t+1h}$, $A_{t+2h}$, and $A_{t+3h}$. Each attention map is produced by 2D convolution layers with LeakyReLU as their activation function (Fig. 4 and Table 2). The first attention map $A_{t+1h}$ takes a sigmoid-transformed $y_{t-0\text{min}}$, the latest available rain map, as its input. Each of the other two attention maps takes its 60-min-earlier attention map as its input. To generate the final prediction $\hat{Y}_t$ at time $t$, attention map $A_t$ is then pixelwise multiplied with the corresponding output of the forecaster $F_t$ to yield the final prediction. That is,

$$\hat{Y}_t = F_t \ast A_t, \quad \text{for } t \in \{t+1\ h, t+2\ h, t+3\ h\},$$

where the asterisk denotes pixelwise multiplication.

d. Discriminator module

As mentioned in section 1, the proposed discriminator module learns to distinguish between actual rain maps and generated rain maps. The discriminator is integrated just the same way as it is integrated into a GAN setup. There are two optimizers. One optimizer optimizes the weights of the encoder and forecaster. The other optimizer optimizes the weights of the discriminator. These two optimizers are executed alternatively. We also note that the discriminator module is not used during the evaluation (prediction) phase.

Consequently, the loss from the discriminator nudges the model toward generating realistic rain maps. We observe that it is especially important for the predictions in the second and the third hour, where the deep learning model tends to be uncertain about its predictions. After experimenting with different architectures, we see that a discriminator formed with simple dense layers (Table 3) is sufficient for this purpose. Combining the encoder–forecaster, attention, and discriminator modules leads to a novel deep learning model proposed by us (Fig. 3).

e. Loss functions

In the proposed model (Fig. 3), prediction and discriminator losses were calculated and used simultaneously to update the model weights.

Prediction loss is described here. To improve the WMAE metric illustrated in Eq. (1), we follow Shi et al. (2017) to adopt a similar weighted loss function of the same form with
The “state kernel” indicates the state-to-state kernel size in the RNN block, and the dilation size of state-to-state convolution is 1 × 1 for all ConvGRUs. Except for fConv0, all Conv and Deconv layers are followed by a LeakyReLU with a negative slope = 0.2. “CH I/O” denotes the input/output channels of the layer. “In res” and “out res” denote the resolution of input and output, respectively. “In” specifies from which layer the current layer takes its output. “In_state” denotes the initial state of the RNN module.

**Table 1. Neural network configurations of the encoder–forecaster module.** The kernel size and stride of each layer are listed. The padding for all layer types (Conv, ConvGRU, and Deconv) is 1 × 1. The “state kernel” indicates the state-to-state kernel size in the RNN block, and the dilation size of state-to-state convolution is 1 × 1 for all ConvGRUs. Except for fConv0, all Conv and Deconv layers are followed by a LeakyReLU with a negative slope = 0.2. “CH I/O” denotes the input/output channels of the layer. “In res” and “out res” denote the resolution of input and output, respectively. “In” specifies from which layer the current layer takes its output. “In_state” denotes the initial state of the RNN module.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Kernel size</th>
<th>Stride</th>
<th>State kernel</th>
<th>CH I/O</th>
<th>In res</th>
<th>Out res</th>
<th>In</th>
<th>In_state</th>
</tr>
</thead>
<tbody>
<tr>
<td>eConv1</td>
<td>Conv</td>
<td>7 × 7</td>
<td>5 × 5</td>
<td>—</td>
<td>2/8</td>
<td>540 × 420</td>
<td>108 × 84</td>
<td>In</td>
<td>—</td>
</tr>
<tr>
<td>eRnn1</td>
<td>ConvGRU</td>
<td>3 × 3</td>
<td>1 × 1</td>
<td>5 × 5</td>
<td>8/64</td>
<td>108 × 84</td>
<td>108 × 84</td>
<td>Conv1</td>
<td>—</td>
</tr>
<tr>
<td>eConv2</td>
<td>Conv</td>
<td>5 × 5</td>
<td>3 × 3</td>
<td>—</td>
<td>64/192</td>
<td>108 × 84</td>
<td>36 × 28</td>
<td>Rnn1</td>
<td>—</td>
</tr>
<tr>
<td>eRnn2</td>
<td>ConvGRU</td>
<td>3 × 3</td>
<td>1 × 1</td>
<td>5 × 5</td>
<td>192/192</td>
<td>36 × 28</td>
<td>36 × 28</td>
<td>Conv2</td>
<td>—</td>
</tr>
<tr>
<td>eConv3</td>
<td>Conv</td>
<td>3 × 3</td>
<td>2 × 2</td>
<td>—</td>
<td>192/192</td>
<td>36 × 28</td>
<td>18 × 14</td>
<td>Rnn2</td>
<td>—</td>
</tr>
<tr>
<td>eRnn3</td>
<td>ConvGRU</td>
<td>3 × 3</td>
<td>1 × 1</td>
<td>3 × 3</td>
<td>192/192</td>
<td>18 × 14</td>
<td>18 × 14</td>
<td>Conv3</td>
<td>—</td>
</tr>
<tr>
<td>fRnn3</td>
<td>ConvGRU</td>
<td>3 × 3</td>
<td>1 × 1</td>
<td>3 × 3</td>
<td>192/192</td>
<td>18 × 14</td>
<td>18 × 14</td>
<td>Rnn3</td>
<td>eRnn3</td>
</tr>
<tr>
<td>fDeconv3</td>
<td>Deconv</td>
<td>4 × 4</td>
<td>2 × 2</td>
<td>—</td>
<td>192/192</td>
<td>18 × 14</td>
<td>36 × 28</td>
<td>Rnn3</td>
<td>—</td>
</tr>
<tr>
<td>fRnn2</td>
<td>ConvGRU</td>
<td>3 × 3</td>
<td>1 × 1</td>
<td>5 × 5</td>
<td>192/192</td>
<td>36 × 28</td>
<td>36 × 28</td>
<td>Deconv3</td>
<td>eRnn2</td>
</tr>
<tr>
<td>fDeconv2</td>
<td>Deconv</td>
<td>5 × 5</td>
<td>3 × 3</td>
<td>—</td>
<td>192/64</td>
<td>36 × 28</td>
<td>108 × 84</td>
<td>Rnn2</td>
<td>—</td>
</tr>
<tr>
<td>fRnn1</td>
<td>ConvGRU</td>
<td>3 × 3</td>
<td>1 × 1</td>
<td>5 × 5</td>
<td>64/64</td>
<td>108 × 84</td>
<td>108 × 84</td>
<td>Deconv2</td>
<td>eRnn1</td>
</tr>
<tr>
<td>fDeconv1</td>
<td>Deconv</td>
<td>7 × 7</td>
<td>5 × 5</td>
<td>—</td>
<td>64/8</td>
<td>108 × 84</td>
<td>540 × 420</td>
<td>Rnn1</td>
<td>—</td>
</tr>
<tr>
<td>fConv1</td>
<td>Conv</td>
<td>3 × 3</td>
<td>1 × 1</td>
<td>—</td>
<td>8/8</td>
<td>540 × 420</td>
<td>540 × 420</td>
<td>Deconv1</td>
<td>—</td>
</tr>
<tr>
<td>fConv0</td>
<td>Conv</td>
<td>1 × 1</td>
<td>1 × 1</td>
<td>—</td>
<td>8/1</td>
<td>540 × 420</td>
<td>540 × 420</td>
<td>Conv1</td>
<td>—</td>
</tr>
</tbody>
</table>

Eq. (2) as the weighting function. We also conducted an extensive study on the influence of the loss function and designed a weighted mean-squared error (WMSE) version that replaces the absolute error in Eq. (1) with the squared error. In the appendix, we show that using WMAE over WMSE is better specifically for low to moderate rainfall regions. The discriminator loss is described as follows. From Eqs. (1) and (2), the predictions on pixels with rain rates less than 0.5 are ignored. We argue that this is the root cause of haziness (especially in later prediction hours) discussed in the case studies (section 5). The design of our discriminator module is used to combat this issue. The discriminator module itself is trained with a cross-entropy loss that distinguishes $\hat{Y}_t$ from $Y_t$.

$$L_D = -\sum_t \log[D(Y_t)] + \log[1 - D(\hat{Y}_t)].$$

Next, another loss component $L_{GD}$ is introduced, which updates the attention and encoder–forecaster weights to generate predictions that could fool the discriminator into thinking of them as real rain maps. Namely, their loss is

$$L_{GD} = -\sum_t \log[D(\hat{Y}_t)].$$

We take a weighted average of the prediction loss and $L_{GD}$ as the loss expression for the nondiscriminator modules. That is,

$$L_{GD} = (1 - \alpha)L_P + \alpha L_{GD},$$

where $L_P$ is the WMAE prediction loss and $\alpha$ is the balancing parameter.

**f. Training details and model hyperparameters**

The code is written in Python using Pytorch Lightning==1.0.2 and Pytorch==1.6.0. We used Adam as the optimizer with 0.0001 as the learning rate. For all experiments, the batch size is set to 16, and we train for a maximum of 15 epochs. Notably, the number of epochs needed should depend on the amount of data present in a single training sample and the number of samples that constitute all batches in one epoch. Since we are working with six frames of 540 × 420 resolution as input, an enormous 1.3 million pixel value for each rain and radar datum, the model gets trained within 15 epochs. In addition, we pick the best-performing model on validation data using the WMAE as in Eq. (1). Since the total number of epochs (15) is low, we did not feel the need to explicitly use early stopping. Also, we did not empirically find any benefit of using dropout.

For memory and file system management, we converted the rain and radar values to integer values and normalized the rain and radar data by dividing them by their respective 95th-quantile value. Note that this normalization is done for creating input data for the model. The target rain maps do not undergo normalization.

Next, we lay out the configuration of our discriminator and attention modules. First, the discriminator model comprises three dense layers with sizes 128, 128, and 1. LeakyRelu is
used for nonlinearity after each dense layer except for the last layer, where a sigmoid is used (Table 3). Second, the attention module comprises five convolution layers, whose output channel counts are [16, 32, 32, 32, 1], kernel size is $5 \times 5$, and padding used is $2 \times 2$. LeakyReLU is used for nonlinearity after every convolution layer except the last one, where a sigmoid function is used (Table 2). Last, $\alpha = 0.02$ is used for computing $L_C$. We obtained the architectures and hyperparameters after fine-tuning on the validation set.

4. Statistical evaluation of model performance

This section presents the results of multiple variants of our model to showcase the benefit of different components (i.e., discriminator and attention module). The baseline model, named the GRU+WMAE model, only contains the encoder–forecaster module and uses WMAE as the total loss. Clear and attentive precipitation nowcasting (CAPN), the best model in this study, has both attention and discriminator modules. The clear precipitation nowcasting (CPN) model has the discriminator but without the attention module. In this case, the output of the encoder–forecaster is the final prediction. In addition, we also developed the balanced precipitation nowcasting (BPN) model, which has neither attention nor discriminator modules but focuses on generating clear predictions by modifying the loss function. It serves as a benchmark for comparing clarity. The idea of BPN is to account for pixels that are not considered in Eq. (2) (i.e., pixels with hourly rainfall < 0.5 mm). Let a mask $M_t$ be a $[0, 1]$ valued tensor of the same shape as $Y_t$ and $M_t[j, k] = 1$ if and only if $Y_t[j, k] < 0.5$. Balanced loss component $L_{bal}$ is defined as

$$L_{bal} = \frac{1}{CN_{time} \times K} \sum_{t, i, j, k} M_t[j, k] \times |Y_t[j, k] - \hat{Y}_t[j, k]|.$$  \hspace{1cm} (11)

The loss for optimizing the BPN model is $(1 - \omega_{bal}) \times L_P + \omega_{bal} \times L_{bal}$. It is easy to observe that the $L_{bal}$ component of the loss penalizes prediction errors only on low-rainfall pixels and pixels without rainfall. Hence, we expect and also observe that the BPN model leads to clearer predictions. Here, $\omega_{bal} = 0.01$ is used, which is obtained by looking at the validation set performance on the WMAE metric.

We use multiple metrics, including CSI, HSS, and performance diagram, to evaluate the performance of our model quantitatively and show a case study for qualitative evaluation. For both CSI and HSS, the higher the value is, the better is the model. Also, we evaluate the performance on three date ranges: (i) the entire year of 2018; (ii) April–September of 2018, the rainy season; and (iii) the rest of 2018 (January–March and October–December).

For benchmarks, the QPESUMS extrapolation technique is used; it is currently used in CWB operation, but the effective forecast time of this technique is only up to one hour. As the values of both CSI and HSS can easily change due to the sample selection of different domain sizes, events, and periods, it is worth noting that the QPESUMS extrapolation technique is usually considered skillful for 0–1-h rainfall prediction under 20 mm in Taiwan. Therefore, the readers should keep in mind that the CSI or HSS of the 20-mm threshold associated with the QPESUMS extrapolation technique can be taken as an acceptable value of the metric. In addition, as another benchmark, we use the latest rain-rate map as prediction, which we refer to by “Persist_10 min.” We also retrain the representative TrajGRU model of Shi et al. (2017) with our dataset for comparison.

The HSS and CSI of the first-hour prediction for various rainfall thresholds are shown in Fig. 5. For all thresholds greater than 5 mm h$^{-1}$, we observe that the CAPN (Fig. 5, purple line) performs best in both cases. The benefit of predicting heavy rainfall is significant if one compares it with the QPESUMS extrapolation technique (Fig. 5, pink line) or the TrajGRU model (Fig. 5, blue line). For >5 mm h$^{-1}$ rainfall, CAPN has HSS in the range 0.3–0.5 and CSI in the range 0.2–0.35, which is significantly better than that of QPESUMS and Persist_10 min. Furthermore, the HSSs/CSIs for different seasons of 2018 show a consistent trend.

We next focus on performance due to individual components. Our claim that the discriminator-based approach gets relatively clear predictions can be inferred by observing the CPN model’s superior performance on thresholds of 1 and 2 mm h$^{-1}$ with respect to models not having the discriminator (all except CAPN and CPN in Fig. 5). We developed the BPN model specifically to compare clarity in prediction obtained by using a discriminator and found that the CPN substantially outperforms the BPN model on small thresholds. Although the benefit of the attention module cannot be exclusively observed by comparing the HSSs/CSIs of the CAPN model with the CPN model (Fig. 5), the case study shown later in section 5 reveals that the CAPN model has a better ability to generate small-scale convective features, especially in the second- and third-hour prediction. In the appendix, we further show that attention improves the performance of the worst-performing pixels during heavy-rainfall cases. We observe a decent performance for small

\begin{table}
\begin{tabular}{|c|c|c|c|c|}
\hline
Type & Kernel size & Stride & Shape input & Shape output \\
\hline
MaxPool2d & 5 × 5 & 5 × 5 & 1 × 540 × 420 & 1 × 108 × 84 \\
Linear & — & — & 9072 & 128 \\
Linear & — & — & 128 & 128 \\
Linear & — & — & 128 & 1 \\
\hline
\end{tabular}
\end{table}
rainfall on HSSs/CSIs of the Persist\_10min and the QPESUMS. It is presumably the case for nonmoving rainfall occurrences and low thresholds. It is so because, for low thresholds, the model need not account for changes in the intensity of the rainfall with time. In the appendix, we show that it is relatively easy to get a model that performs better than Persist\_10min on low-rainfall thresholds, and, to that end, we develop a blended model that combines the predictions of CAPN and Persist\_10min.

For the prediction of 1–2- and 2–3-h rainfall (Fig. 6), as expected, the HSS scores are decreased dramatically. Taking the CAPN as an example (Fig. 6, purple line), the performance...
Fig. 6. HSS scores of the (left) second-hour and (right) third-hour rainfall prediction in (a),(b) 2018; (c),(d) the 2018 rainy season; and (e),(f) the 2018 cold season for the TrajGRU (Shi et al. 2017), GRU+WMAE, BPN, CPN, and CAPN models for various rainfall thresholds. The HSS scores of the first-hour prediction for TrajGRU (blue dashed line) and CAPN (purple dashed line) are also displayed.
for thresholds under 20 mm of the 1–2-h prediction is only about 50% when compared with that of the 0–1-h prediction (Fig. 6, purple dashed line). Also, the models lose skills significantly to predict rainfall > 20 and > 10 mm in 1–2- and 2–3-h, respectively. Nevertheless, as revealed later by the case studies, the capability of the CAPN model to predict small-to-medium hourly rainfall roughly still helps in providing 0–3-h QPN.

5. Case studies

a. The frontal rainfall case on 7 May 2018

A case study of the frontal system near Taiwan on 7 May 2018 illustrates the performance of QPN models developed in this study (Figs. 7–9). As shown in Fig. 7, any target can be the first-hour, second-hour, or third-hour prediction by different predicting initial times. Specifically, for predicting hourly rainfall for 1700–1800 UTC as the second-hour prediction, input data would be from two hours ago (1500–1600 UTC). The models predict three hourly rain maps (1600–1700, 1700–1800, and 1800–1900 UTC), and Fig. 7 displays all three predictions for each target with three rows (1 h before, 2 h before, and 3 h before).

The operational QPESUMS radar echo extrapolation (Fig. 7, label “A”) fairly captures the motion of the frontal rainband in the first hour but overpredicts the maximum rainfall to more than 70 mm while the maximum rainfall in the target observation is around 40 mm. The baseline GRU+WMAE model captures the rainband movement and the maximum rainfall; however, it overpredicts the light rain, leading to unrealistic large raining areas and reducing the forecasters’ trust in the model (Fig. 7, “C”).

On the other hand, for the advanced deep learning QPN models, the BPN, CPN, and CAPN visually exhibit comparable 0–1-h predictions (Fig. 7, “B”). Both the BPN model and the CPN model solve the problem of over forecasting the light rain, but the CPN better generates the real-like features with a discriminator module. Moreover, the CAPN model performs better for the second and third-hour predictions by adding the attention module into the model. Specifically, as seen in Fig. 7 (“D”) and Fig. 7 (“E”) with red bounding boxes, the BPN and the CPN models fail to capture the detailed convective structures of the frontal rainband in the 1–2- and 2–3-h predictions while the frontal system is much more identifiable in the CAPN model. The CAPN model retains the frontal rainband convection (Fig. 7, “F”) with the help of the attention maps. Although the accuracy of rainfall maximum in the 2–3-h prediction is lower than that of 1–2-h, the CAPN model is still helpful for anticipating the heavy-rain areas as the convective structure is still recognizable even for the third-hour prediction. Therefore, the CAPN model helps greatly due to the lack of this kind of product (i.e., the 1–2- and 2–3-h rain maps) at present, which are generated only based on observation. Although NWP models can provide hourly rain maps, they usually suffer from the imperfect model initial fields (and ineffective data assimilation in the model spinup period) and
could not be provided to the forecasters in time, particularly for 0–3-h nowcasting.

The performance diagrams of this case (Fig. 8) suggest that, for the 0–1-h prediction (Fig. 8a), the deep learning models outperform the operational QPESUMS extrapolation technique for every rainfall threshold greater than 5 mm. Specifically, the deep learning models have higher CSI scores (Fig. 8a, green contours) than the QPESUMS. Furthermore, for 1-, 3-, and 5-mm light rain thresholds, our modifications in the model (i.e., balanced loss, discriminator, and attention) gradually improve the CSI by increasing the success ratio (SR; Fig. 8a, x axis) and keeping the probability of detection (POD; Fig. 8a, y axis) high. Most important, for the higher 20–40-mm thresholds, the CAPN model outperforms other models because of its higher PODs. The threshold of 40 mm of rainfall per hour is also the criterion of torrential rainfall formulated by CWB, so the CAPN model demonstrates practicality in operational forecasting. Note also that, with the rainfall threshold increasing, the QPESUMS extrapolation technique has simultaneously decreased PODs and SRs because the extrapolation generally keeps the distribution of the rain rates at the initial time of the forecast. In contrast, the deep learning models tend to overpredict when considering smaller thresholds (high POD and lower SR) and underpredict when considering larger thresholds (high SR and lower POD).

For the 1–2- and 2–3-h verification, the performance diagrams (Figs. 8b,c) reveal significantly decreased CSIs. Nevertheless, the CAPN model is better than CPN and shows some skills for the second-hour prediction and has CSIs of ∼0.23 and ∼0.20 for 15- and 20-mm thresholds, respectively (Fig. 8b); these CSIs suggest comparable performance to the 0–1-h prediction of the QPESUMS extrapolation. As for the 2–3-h prediction, the relative usable rainfall range declines to 1–10 mm, which still needs improvement.

This study also evaluates the model performance by verifying 0–3-h accumulated rainfall (Figs. 9 and 8d). The rain maps of 0–3-h rainfall clearly show that the baseline GRU+WMAE
model overpredicts in no and light rain areas (Fig. 9a). Note that the upgraded models (BPN, CPN, and CAPN) fix this issue. The CAPN model also generates more real-like details, like the strong convective spots of approximately 70–90 mm dispersing in the rainband, although the locations of rainfall extremes are slightly shifted. On the other hand, the performance diagram (Fig. 8d) shows that, for 3-h accumulated rainfall, the upgraded models outperform the baseline GRU+WMAE model for multiple thresholds. Note also that the CAPN model tends to have higher PODs (Fig. 8d, y axis) for 30–50-mm thresholds and could be thus more useful for disaster prevention. In summary, we suggested that the CAPN model is useful and valuable for the forecasters because it generates proper raining areas, captures the medium-to-high rainfall, and even hints at convective extremes.

b. An afternoon thunderstorm case on 4 June 2021

This section provides another case study to showcase the capability and usefulness of the CAPN model for predicting extreme rainfall events (Fig. 10). On 4 June 2021, the atmospheric environment around Taiwan was very unstable due to a front located north of Taiwan and a tropical cyclone in the South China Sea (Fig. 10b). The low-level background flow affecting Taiwan is from south to southeast, suitable for severe afternoon thunderstorms developing in the Taipei Basin. Figure 10c (black-outlined box) shows that the daily accumulated rainfall is over 200 mm in downtown Taipei, causing serious flash floods (Fig. 10a). Also, the accumulated value of the rainfall and the intense rain rates (Fig. 10d, black dashed line and bars) reveal a high probability of flooding in Taipei.
A series of CAPN predictions are examined to evaluate the capability of CAPN to predict the development of afternoon thunderstorms (Figs. 10d and 11). From an intuitive point of view, the CAPN model can provide rainfall predictions close to the ground truth (Figs. 11b,c). Comparing the last input rain rates of the model (Fig. 11a) with the 0–1-h prediction (Fig. 11c), the CAPN model captures the initialization of the thunderstorm in northern Taiwan at 0330 UTC and is capable of predicting the rainfall maxima of the developing storm for the first-hour predictions after 0400 UTC (Fig. 11b,c). Note that the first row of Fig. 12 shows the rain rates in millimeters per hour, so the 0–1-h rainfall prediction (Fig. 11c, mm) does imply the developing convective cells and shows higher rainfall values than the input (Fig. 11a). Furthermore, the comparison between the target 0–3-h rainfall (Fig. 11d) and the CAPN 0–3-h prediction (Fig. 11e) suggests that the CAPN model effectively alerts the maxima of the future 3-h accumulated rainfall. For example, the prediction made at 0430 UTC (Fig. 11, central column) captures the primary rainfall extreme in northern Taiwan and secondary local rainfall maxima in central Taiwan. Furthermore, as shown in Fig. 10d, the brown lines representing the 0–3-h CAPN forecasts every hour suggest that the CAPN model anticipates the occurrence of heavy rainfall at 0600 UTC and fairly predicts the maximum accumulated rainfall value at 0800 UTC, about 2 h ahead of the accumulated rainfall peak.

We plot the performance diagrams (Fig. 12) of this thunderstorm case as in Fig. 9, except that the scale-aware scores of different rainfall thresholds are plotted. The scale-aware CSI, POD, and SR were calculated based on an alternative definition of binary classification: for an individual grid, if the maximum rainfall within the area of interest (e.g., 5-km box, 12.5-km box, or 25-km box) is larger than the given rainfall threshold, this grid is then classified as true, and vice versa. This idea is pretty similar to the fractions skill score (FSS; Roberts and Lean 2008; Roberts 2008) but is easy to calculate and display on a performance diagram. In this manner, we can discuss the scores, which are more relevant to the decision-making processes, on the village (5 km), city (12.5 km), and county (25 km) scales. The results show that, for the 0–1-h prediction of 40-mm rainfall (an important threshold in CWB operation), the CAPN model has CSIs of ~0.25, ~0.29, and ~0.36 at the village, city, and county scales, respectively. Also, Fig. 12b suggests that the CAPN model may have the skills to predict 3-h rainfall > 50–70 mm, as the CSIs are larger than 0.25. Here, we argue that CSI > 0.25 is considered to be skillful for rainfall prediction because the QPESUMS extrapolation technique usually has CSI ≈ 0.2 for
20-mm hourly rainfall for the 0–1-h prediction, and this performance is considered useful during the CWB nowcasting operation. To sum up, it is suggested that our new model can provide an effective warning of urban-area torrential rainfall for afternoon-thunderstorm cases, which is critical to urban flash flood prevention.

6. Concluding remarks and future work

In this study, deep learning QPN models are developed for providing high-resolution precipitation forecasts for up to three hours around Taiwan and the surrounding area. These models use inputs of column-maximum radar reflectivity and rain rates every 10 min from the last hour and predict the hourly accumulated rainfall in the next three hours. The proposed deep QPN model is based on the convolutional recurrent neural network with GRU blocks and has a unique design to integrate discrimination and attention techniques. This design allows our CAPN model to address two important demands: (i) to produce predictions that meteorologists can trust in both low- and high-rainfall regions and (ii) to improve the prediction quality for longer periods.

In many recent works that aim to predict high-rainfall events accurately, the model usually produces blurry rain-map nowcasts that overestimate in low-rainfall regions (i.e., drizzle everywhere). Such nowcasts are visually less trustworthy and practically less useful for forecasters. Therefore, a discriminator is employed to fix the trust issue. The discriminator module in the CPN and CAPN models encourages the encoder–forecaster module to generate clearer and more realistic rain maps without sacrificing the predictive accuracy of extreme rainfall. The statistical verification based on the testing dataset (section 4b) suggests that adding the discriminator module (the CPN model) contributes to a considerable increase in CSIs and HSSs for the hourly rainfall thresholds less than 2 mm, when compared with the WMAE+GRU baseline model.
We further improved the performance at thresholds larger than 5 mm by dynamically rescaling the prediction using an attention module, making the CAPN model capable of generating more meso-β-scale rainfall features. The consecutive attention across different hours also helps to extend the nowcasting time frame from typically 0–1 to 0–3 h, further addressing the needs for socioeconomic weather-dependent decision-making. Note that providing a reliable 0–3-h prediction is preferred by forecasters over short 0–1-h predictions as done in many previous studies (Shi et al. 2017; Tran and Song 2019; Franch et al. 2020; Ayzel et al. 2020; Trebing et al. 2021).

In addition to the statistical verifications, two case studies also demonstrate that the CAPN model can capture the motion of rainbands in a frontal case and provide an effective warning of urban-area torrential rainfall in another afternoon-thunderstorm case. For the frontal case, the models with the discriminator (CPN and CAPN) do not generate predictions that rain everywhere and therefore do not overpredict light rain. Verifying 0–3-h accumulated rainfall further shows that the upgraded models fix this issue, whereas the baseline GRU+WMAE model overpredicts in the no rain and light rain area. Furthermore, the CAPN model shows its capability to generate good rain areas, capture medium-to-high rainfall, and even provide hints of convective extremes, especially in the second- and third-hour predictions. For the afternoon-thunderstorm case, the CAPN model could effectively warn of urban-area torrential rainfall critical to urban flash floods; it captures the initialization of the thunderstorm and can predict the rainfall maxima of the developing storm.

In summary, the key idea of this study is that if a CNN-RNN model is upgraded with proper modules (attention and discriminator), we can expect that a data-driven nowcasting model provides clear rain maps and extends the forecast period to three hours. Although this may greatly help forecasters, this kind of data-driven QPN model still has great potential, which is suggested to be explored in future studies. Last, we acknowledge the limitations of this study that we plan to take up in the future. A well-trained human weather forecaster typically combines guidance based on multiple observations, analyses, statistical techniques, and NWP models to provide his final QPN. Taking the afternoon thunderstorms in Taipei as the case, the maintenance of the convective system can be affected by the vertical environmental thermodynamic profile and the ventilation of middle-tropospheric flow. Also, the development of sea-breeze circulation revealed by surface station observation and the shallow-cloud development observed by visible-channel satellite images could be critical for anticipating convective initialization. Note that this kind of information is not contained in the rain and radar data used in this study. Therefore, how a deep learning QPN model better uses multiple and heterogeneous data is a key objective of our future study.

Acknowledgments. The authors appreciate the feedback from CWB forecasters and the constructive reviews of the three anonymous reviewers. Computing resources for this study were mainly provided by the Center for Weather Climate and Disaster Research, National Taiwan University. This work was funded by Grants 110-2111-M-002-016, 111-2111-M-002-016, and MOST 110-2628-E-002-013 of the Ministry of Science and Technology, Taiwan, and Project 1112063A of the Central Weather Bureau, Taiwan.

Data availability statement. CWB Weather Forecast Center provided the QPESUMS raw data used in this study. Because of confidentiality agreements, supporting data can only be made available to bona fide researchers subject to a
nondisclosure agreement. Details of the data and how to request access are available from Dr. Treng-Shi Huang at CWB. However, samples of the compiled dataset for model training and validation and the code are available on Github (https://github.com/ashesh-0/AccClearQPN).

APPENDIX

Ablation Studies on Loss Function, Attention, and Insights on Further Improvement

a. Comparing MSE versus MAE as loss function

A test on the loss functions to understand the behavior of MAE and MSE is conducted. We designed an WMSE loss, which replaces the mean absolute error in Eq. (1). We also tried with the average loss of WMAE and WMSE as the final loss. As shown in Fig. A1, the model trained with the WMSE loss is significantly inferior in lower to moderate rainfall cases. The result is intuitive since WMSE gives much more importance to larger errors, typically with heavy rainfall. Thus, a lesser emphasis on lower errors (and lower rainfall cases) leads to this effect. We consequently chose the weighted MAE as the loss function in this study.

b. Understanding where attention has a larger impact

It is naturally motivating to know whether there exist specific cases where the attention module enhances the prediction performance. We found that the attention module improves the prediction of the worst-performing pixels in heavy-rainfall cases (Fig. A2). For every target frame with 226,800 pixels, we

![Graphs showing HSS and CSI for the GRU+WMSE, GRU+WMAE, and GRU+WMAE&WMSE models. One can observe that MSE component in the loss leads to inferior performance for all thresholds less than 20 mm.](image)

**Fig. A1.** (a) HSS and (b) CSI for the GRU+WMSE, GRU+WMAE, and GRU+WMAE&WMSE models. One can observe that MSE component in the loss leads to inferior performance for all thresholds less than 20 mm.

![Bar charts comparing absolute error in prediction on the worst-performing pixels. The average of percentile errors on August 2018, the month with the highest rainfall, is computed. For reference, 227 pixels have worse performance than the 99.9th-percentile value. Note that the mean and standard errors computed using five instances of each model are shown. The number shown above bar pairs is the value difference; H0, H1, and H2 represent the first-, second-, and third-hour predictions, respectively.](image)

**Fig. A2.** Comparing absolute error in prediction on the worst-performing pixels. The average of percentile errors on August 2018, the month with the highest rainfall, is computed. For reference, 227 pixels have worse performance than the 99.9th-percentile value. Note that the mean and standard errors computed using five instances of each model are shown. The number shown above bar pairs is the value difference; H0, H1, and H2 represent the first-, second-, and third-hour predictions, respectively.
compute the pixelwise absolute error between the target and the prediction on these pixels and fetch 99.9th, 99.99th, and 100th-percentile values. Performance on these three percentiles is shown separately in the three subplots of Fig. A2. Results show that the CAPN model is better in terms of mean and standard error (we have trained five models for each configuration). Therefore, besides the overall performance improvement, the attention module enhances the predictive stability as the worst-pixel performance is improved.

c. Blending model: Improving prediction on light rain area

Beyond clarity, which our model achieves, meteorologists are not concerned with very low rain-rate thresholds. Nonetheless, we argue that if needed, it is relatively easy to get even better performance on lower thresholds with a blended version of our model, particularly for the first-hour prediction. A blended version of our model is showcased here. We first create a classification model that predicts whether a pixel will have an hourly rainfall exceeding 0.5 mm, and our final prediction for the blended version is the weighted average of prediction of CAPN model and last 10-min rain rate. The weights are computed using the classifier’s prediction and ensure that the CAPN model’s prediction gets more weight when a higher rain rate is expected. As shown in Fig. A3, the blended model outperforms the Persist_10min benchmark on lower thresholds and achieves similar performance to our best CAPN model on higher thresholds. However, we do not prefer the blending, since, similar to the Persist_10min benchmark, it is inferior at later hours.

REFERENCES


Franch, G., D. Nerini, M. Pensesini, L. Coviello, G. Jurman, and C. Furlanello, 2020: Precipitation nowcasting with orographic enhanced stacked generalization: Improving deep learning...


Vondrick, C., H. Pirsiavash, and A. Torralba, 2016: Generating videos with scene dynamics. 30th *Int. Conf. on Neural Information Processing Systems*, Barcelona, Spain, Association for Computing Machinery, 613–621.

