Authors’ responses (GMD-2022-276)

The authors would like to thank the editor for your precious time and invaluable comments. The corresponding changes and refinements are highlighted in yellow in the revised paper and are also summarized in our responses below. Authors’ responses are in blue. Editor’s comments are in black. When the manuscript in cited, it is shown in *italics*.

Response to RC2

This paper compares different ML configurations for precipitation nowcasting. All the ML algorithms compared are deep learning (DL) algorithms. The aim of the paper is to figure out the best configuration and provide some guidelines for the best modeling strategy. I think this goal project is somewhat ill-defined just because there are so many different choices available, and the results of the study are quite inconclusive (discussed below in specific comments); but still it is important to explore these choices systematically as done in the paper.

➔ The authors sincerely appreciate your valuable time and effort in providing constructive comments. Based on your suggestions, we have conducted intensive modeling and analysis, which took more time and effort than our original manuscript. By striving to address the suggested comments appropriately, we believe that this paper has been significantly improved in this time, resulting in more robust findings.

There are some major issues with the methodology followed in the paper. The DL models trained using the ADAM optimizer can show very different performances based on random initialization – this is what I have seen in the streamflow simulation applications. Therefore, many different models must be trained using different initializations (i.e., different random seeds); an average of the models can be taken as the final model. It does not seem like the authors have followed this procedure; in which case, the results reported are unlikely to be robust. So, this needs to be taken care of before the paper can be considered for publication.

➔ As you pointed out, the randomness should be considered in DL-QPN to assure robust prediction. We overlooked this in many previous studies. Based on your comment, we ran each scheme five times and aggregated them to generate the final results with random seeds of 0, 999, 2023, 44919, and 2022276. With 12 initial schemes, a total of 60 runs were conducted, which was very time and resource-intensive. Since the number of runs in DL training increased fivefold, we changed the training samples from all seasons to only summers and increased the total period from 2011-2020 to

2012-2022 by adding newly collected data. This was described in lines 243-257 and Table 2.

(Lines 243-257)

In DL models, there are various aspects that contain randomness, such as random weight initialization or a random mini batch from the entire samples. To evaluate the model performance more reliably, we ran each scheme 5 times with different random seeds of 0, 999, 2023, 44919, and 2022276. Starting from combining DL models (U-Net or ConvLSTM), loss functions (MSE or BMSE), and input sequence length (1 h, 2 h, or 3 h), total of 12 initial schemes were generated before ensemble averaging. As each initial scheme had 5 runs with different random scenes, a total of 60 runs for training DL models were conducted in this study (Table 2 and Figure 4). By aggregating 5 members for each scheme, each scheme can assure more stability than a single run. For example, UM1 stands for the average of U-Net trained by MSE loss with an input sequence of 1 h for 5 different random seeds. The average of each scheme with different input sequence lengths was again aggregated; for example, UM is the ensemble mean of UM1, UM2, and UM3. Consequently, UM is the mean of the total 15 runs, same is true for UB, CM, and CB. The remaining ensemble processes are similar to those in Table 2.

Table 2. Specifications of the schemes designed in this study with their abbreviations. 'U' stands for U-Net, 'C' stands for ConvLSTM, and 'P' is for pySTEPS. 'M' and 'B' indicate MSE and BMSE loss functions, respectively. The following number is the length of the input past sequence in hours. The number of runs indicates that there are n total members to yield an average for each scheme with different random seeds to ensure stability and representativeness.

Abbreviation	Model	Loss	Input sequence	# of run
UM1	U-Net	MSE	1h	5
UM2	U-Net	MSE	2h	5
UM3	U-Net	MSE	3h	5
UM	Ensemble averaging (UM1, UM2, UM3)			
UB1	U-Net	BMSE	1h	5
UB2	U-Net	BMSE	2h	5
UB3	U-Net	BMSE	3h	5
UB	Ensemble averaging (UB1, UB2, UB3)			
U	Ensemble averaging (UM, UC)			
CM1	ConvLSTM	MSE	1h	5
CM2	ConvLSTM	MSE	2h	5

CM3	ConvLSTM	MSE	3h	5
CM	Ensemble averaging (CM1, CM2, CM3)			
CB1	ConvLSTM	BMSE	1h	5
CB2	ConvLSTM	BMSE	2h	5
CB3	ConvLSTM	BMSE	3h	5
CB	Ensemble averaging (CB1, CB2, CB3)			
C	Ensemble averaging (CM, CB)			
P1	pySTEPS	-	1h	1
P2	pySTEPS	-	2h	1
P3	pySTEPS	-	3h	1
P	Ensemble averaging (P1, P2, P3)			
UC	Ensemble averaging (U, C)			
UCP	Ensemble averaging (U, C, P)			

- ➔ Based on the idea of aggregation with multiple runs, we extended the experiments of ensemble averaging of different schemes. As the ensemble showed more significant results than initially suggested factors, we added ensemble as one of the four critical key factors. This was described in Section 2.4 and Figure 4.

2.4 Ensemble approach

Ensemble approaches have been adopted as the standard in NWP by combining multiple different members to produce robust results. Despite their potential, ensemble approaches have not been actively adopted in DL-QPN. The ensembling of multiple schemes can bring enhanced and more stable performance than a single model in various evaluation aspects. Ensemble can be executed with simple aggregation with equal weights, or it can employ another machine learning model to learn the optimal way of combining each prediction, such as a stacking ensemble (Cho et al., 2020; Franch et al., 2020). The primary interest of this study is how the results can be improved through ensemble. Hence, only a simple ensemble was done by averaging results from multiple schemes with same weight.

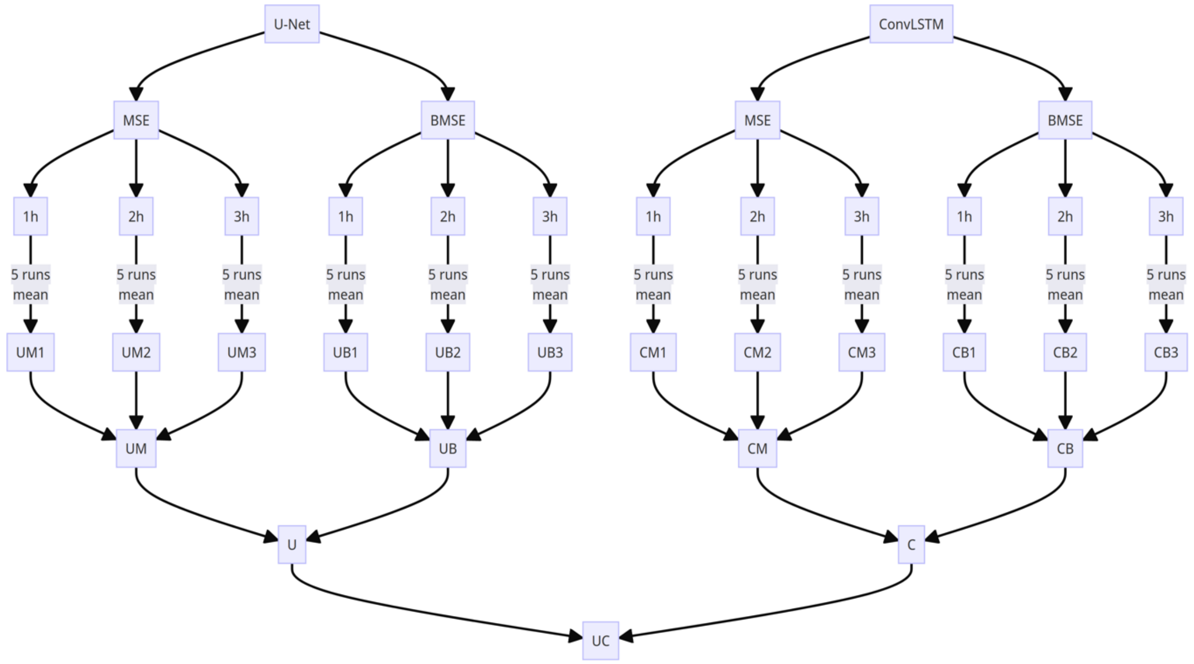


Figure 4. The infographic for overall flowchart in this study. 'U' stands for U-Net, 'C' stands for ConvLSTM, and 'P' is for pySTEPS. 'M' and 'B' indicate MSE and BMSE loss functions, respectively. The following number is the length of the input past sequence in hours.

Further, in the multiple prediction case (MU), one expects that the performance will be worse than the single prediction scenario because in the former the model tries to minimize prediction errors at multiple time-steps. But the advantage of using MU design is that it is trying to capture more information from the data and the synergy between the information at different future time-steps helps improve the prediction. I wonder if it is possible to improve the predictions from the MU by adding small layers at the head of the MU where each layer will be dedicated to a single time-step. This can also be seen as the post-processing of MU predictions. This might be helpful in reducing some of the biases that are shown in Table 4.

➔ When designing the U-Net model for time-series forecasting, the time dimension can generally be treated in the channel axis. In our pilot test, there were several trials to compare the optimal structure among the variants of U-Net. We tested the multi-head structure for different time steps as you suggested, as well as the same structure for each lead time. However, in our preliminary study, we could not find any improvement over the original U-Net structure from RainNet v1.0 (Ayzel et al., 2020). Hence, we adopted the U-Net structure without testing others in our main experiment.

Authors' responses (GMD-2022-276)

The authors have compared their results with the 'persistence' prediction. First of all, persistence has not been defined anywhere in the paper. Assuming that it implies that the forecasted value is the same as at the previous time-step, the reported results are not much better than the persistence (both CSI and MAE). Which raises the question if the DL models are actually extracting any significant information from the data? In this regard, why stop at the previous 120 minutes of past data as input? Why not go further in the past; ConvLSTM could extract useful information from a long past sequence.

- ➔ The missing definition of persistence was also pointed out in RC1. We added its definition in line 282-283.

(Line 282-283)

The *n*-hour persistence model represents a straightforward approach in which the current precipitation is assumed to persist without any change for the next *n* hours.

- ➔ To test various time steps, we extended the length of the input sequence to up to 3 hours. When the maximum lead time is set at 2 hours, three input sequence lengths (i.e., 1h, 2h, and 3h) can show the effect of shorter, same, and longer input lengths compared to output. The updated input sequence is described in the manuscript. Consequently, the scheme configuration was changed, as shown in Table 2.

(Lines 147-149)

To examine the impact of the input sequence length, we compared radar sequences of 1 h, 2 h, and 3 h against 12 future radar scenes for 2 h. This sets up a ratio of input-output sequences at 1:2, 1:1, and 3:2.

- ➔ In the results, there was a weak tendency of performance improvement with a longer input sequence in terms of MSE and Balanced MSE (BMSE) in both 1h (Table 4) and 2h (Table 5) lead times. The effect of input sequence length was covered in the revised manuscript.

Table 4. Quantitative performance over summers of 2020-2022 for the 1 h prediction. Please refer to Table 2 for each scheme. The numbers after metrics indicate the thresholds of precipitation for evaluation. Upward and downward arrows indicate that the metric is better when larger and smaller values are better, respectively.

Scheme	MSE↓	BMSE↓	Bias1	Bias5	Bias10	CSI1↑	CSI5↑	CSI10↑
UM1	10.22	206.43	-0.80	-3.00	-7.93	0.60	0.41	0.22
UM2	9.98	200.15	-0.80	-3.02	-7.96	0.59	0.40	0.22

UM3	**9.73**	192.80	-0.66	-2.75	-7.58	0.59	0.40	0.22
UM	9.97	203.99	-0.80	-3.11	-8.21	**0.61**	**0.42**	0.22
UB1	15.58	151.96	2.62	3.75	3.78	0.51	0.36	0.25
UB2	15.27	148.74	2.52	3.62	3.50	0.49	0.35	0.25
UB3	14.56	150.27	2.22	3.15	2.73	0.48	0.35	0.24
UB	12.87	**144.76**	2.39	3.25	2.44	0.48	0.37	**0.28**
U	10.27	168.64	1.01	0.50	-2.62	0.53	**0.42**	**0.28**
CM1	12.28	251.25	-0.72	-3.87	-11.33	0.51	0.31	0.10
CM2	11.87	242.57	-0.79	-4.09	-11.24	0.51	0.31	0.11
CM3	11.60	235.31	-0.77	-4.14	-11.15	0.51	0.30	0.11
CM	12.09	249.31	-0.80	-4.12	-11.49	0.52	0.32	0.11
CB1	25.52	171.82	4.08	6.15	7.33	0.32	0.23	0.18
CB2	24.67	165.02	3.58	5.99	7.12	0.29	0.23	0.18
CB3	25.57	159.65	3.88	6.25	7.59	0.31	0.23	0.17
CB	24.79	168.68	3.89	6.03	7.11	0.31	0.24	0.18
C	14.79	196.83	2.06	2.42	-0.36	0.37	0.30	0.20
UC	11.72	180.08	1.52	1.21	-2.34	0.43	0.37	0.25
P1	19.18	291.18	-0.19	-0.33	-0.62	0.52	0.35	0.22
P2	18.90	284.38	-0.16	-0.29	-0.55	0.52	0.34	0.22
P3	18.73	279.51	-0.14	-0.26	-0.51	0.52	0.34	0.21
P	16.18	252.12	-0.34	-0.76	-1.41	0.47	0.31	0.19
UCP	11.04	191.04	0.87	-0.08	-4.22	0.44	0.38	0.26
Persistence	21.61	267.60	0.15	0.34	0.55	0.47	0.27	0.16

➔

Authors’ responses (GMD-2022-276)

Table 5. Quantitative performance over summers of 2020-2022 for the 2 h prediction. Please refer to Table 2 for each scheme. The numbers after metrics indicate the thresholds of precipitation for evaluation. Upward and downward arrows indicate that the metric is better when larger and smaller values are better, respectively.

Scheme	MSE↓	BMSE↓	Bias1	Bias5	Bias10	CSI1↑	CSI5↑	CSI10↑
UM1	12.62	257.45	-1.75	-5.63	-13.00	0.49	0.23	0.07
UM2	12.29	249.03	-1.75	-5.59	-12.91	0.48	0.23	0.07
UM3	**12.03**	242.30	-1.63	-5.35	-12.52	0.48	0.23	0.07
UM	12.27	254.84	-1.78	-6.09	-13.69	**0.51**	0.24	0.06
UB1	18.04	208.84	1.80	2.21	0.61	0.38	0.26	0.14
UB2	19.07	196.56	2.31	3.25	2.66	0.39	0.26	0.15
UB3	18.52	197.14	2.12	2.90	1.95	0.39	0.25	0.14
UB	14.83	192.23	1.93	2.30	-0.36	0.37	0.29	**0.17**
U	12.20	218.85	0.50	-1.25	-7.61	0.44	**0.30**	0.13
C1M	13.90	281.72	-1.00	-7.06	-16.73	0.40	0.12	0.00
C2M	13.50	273.02	-1.05	-7.06	-16.53	0.41	0.12	0.00
C3M	13.23	266.75	-1.04	-7.06	-15.95	0.41	0.12	0.01
CM	13.82	281.01	-1.09	-7.22	-16.55	0.41	0.12	0.01
C1B	29.79	198.82	3.05	6.04	6.82	0.16	0.17	0.12
C2B	29.04	190.89	3.01	6.00	6.91	0.16	0.18	0.12
C3B	29.74	**185.57**	3.19	6.23	7.34	0.17	0.17	0.12
CB	29.23	196.31	3.01	5.98	6.78	0.16	0.18	0.13
C	16.99	227.21	1.92	2.16	-3.46	0.27	0.22	0.10
UC	13.64	220.93	1.30	0.18	-7.40	0.33	0.28	0.11
P1	24.33	337.08	-0.31	-0.46	-0.71	0.39	0.21	0.11
P2	23.87	327.89	-0.27	-0.40	-0.62	0.38	0.21	0.11
P3	23.58	321.85	-0.24	-0.37	-0.59	0.38	0.20	0.11
P	19.41	288.13	-0.52	-1.10	-2.03	0.33	0.18	0.10
UPC	13.26	240.00	0.49	-1.69	-9.33	0.35	0.27	0.11
Persistence	25.39	288.91	0.25	0.60	1.03	0.37	0.19	0.10

(Lines 21-22 in Abstract)

There was a small trend where a longer input time (3 h) gave better results in terms of MSE and BMSE, but this effect was less significant than other factors.

(Lines 283-290 in Section 4.1)

A longer input sequence length generally yields lower MSE and BMSE for all schemes, with specific improvements seen for UM and CM in both prediction lead times. This trend can also be observed with UB and CB and, interestingly, in the pySTEPS non-DL model. P3 demonstrated the lowest MSE and BMSE across all P1-P3 schemes in both prediction periods, *suggesting that a longer input sequence can reduce both general capitation (MSE) and high-intensity weighted metrics (BMSE). However, no significant patterns were discernible in other metrics, such as mean bias or CSI, across all schemes with varying input sequence lengths. As the ensemble mean revealed similar results to individual members with different input sequence lengths, subsequent analyses solely focused on aggregated results (i.e., UM, UB, CM, and CB), as illustrated in Figure 5.*

(Lines 500-502 in Conclusion)

In general, there was a weak tendency for longer past sequence lengths to yield lower MSE and BMSE for both DL models, but there was little difference in terms of the CSI.

Specific comments:

Line 70: Why are RNNs unstable? Explain.

➔ In Ayzel et al. (2020), they reported that RNNs are sometimes brittle, and CNNs are more stable, citing Bai et al. (2018) and Gehring et al. (2017). We checked the suggested references and found these supporting parts. The corresponding texts cited references in Ayzel et al. (2020) as follows:

- Bai et al. (2018):
  "However, RNNs have limitations in processing long-term dependencies due to the vanishing gradient problem and are not well-suited for parallelization."
- Gehgring et al. (2017):
  "Compared to recurrent models, computations over all elements can be fully parallelized during training and optimization is easier since the number of non-linearities is fixed and independent of the input length"

➔ As we checked the original references, it seems that they reported the limitations of RNNs as (1) the vanishing gradient problem and (2) difficulty in parallelization. Based on this, we removed this sentence, as it is hard to simply say that "RNN is unstable."

Line 88: Do you want to say the 'sequence length' and 'forecasting length'? Amount of data is usually reserved for sample size.

➔ We updated the sentence to indicate 'sequence length' more clearly.

(Line 87-88)

As the DL-QPN is data-driven, the sequence length will likely determine the model performance

Line 131: A 3x3 kernel was used in this study. Why? Were other kernel sizes tried?

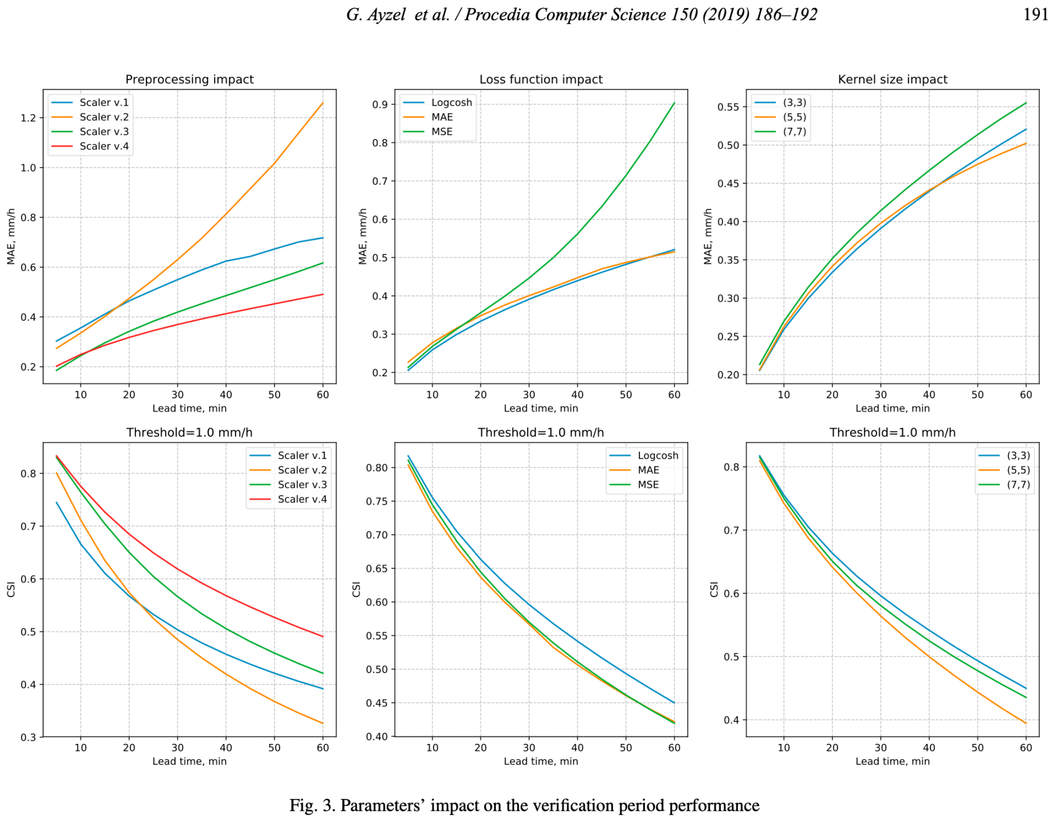
➔ As you pointed out, the size of the kernel can be diverse. However, we set the kernel size to 3x3 for two reasons. (1) As we adopted the U-Net model from RainNet v1.0 (Ayzel et al., 2020), we kept the original kernel size. (2) Prior to Ayzel et al. (2020), Ayzel et al. (2019) explicitly compared the impact of kernel size on the CNN model. They compared kernel sizes of 3x3, 5x5, and 7x7. Contrary to their expectation that increased kernel size may result in better performance, the largest kernel size (7x7) showed the worst result.

(Ayzel et al., 2019)

"4.3. Kernel size impact

Our first guess was that increasing the neural network receptive field by increasing a size of convolutional kernels, would increase the overall network's performance because of accounting of precipitation specific changes from a larger neighborhood. However, the results are different. It is clear that increasing model's receptive field capacity leads to overfitting and increasing uncertainty of predictions. For providing nowcast with the lead time of one hour, we recommend using smaller kernel size of convolutional layers."

Authors’ responses (GMD-2022-276)



G. Ayzel et al. / Procedia Computer Science 150 (2019) 186–192 191

Fig. 3. Parameters’ impact on the verification period performance

Line 148: Vanilla RNNs also have ‘exploding gradient’ problem.

- ➔ Here, "basic RNN" indicates a vanilla RNN. To avoid confusion, we changed "basic RNN" to "vanilla RNN" for clarity.

(Line 127)

As the vanilla RNN structure suffers from the vanishing gradient problem with an increasing number of recurrent hidden layers, revised RNNs, such as long short term memory (LSTM) and gated recurrent units (GRU), have gained widespread acceptance (Cho et al., 2014; Hochreiter and Schmidhuber, 1997).

Line 149: Do you want to say increasing sequence length?

- ➔ No. In this sentence, our intention was to discuss the number of hidden layers of RNN regardless of its sequence length. We updated it according to the previous answer.

Line 161: Why is more diverse input sequence a problem? Yes, there is tradeoff between the calibration sample size and the input sequence length. But did you test what is the optimal sequence length for your models given the amount of available training data?

➔ This sentence is not about the trade-off between sample size and sequence length but simply that a longer sequence can contain both helpful or non-helpful information simultaneously. We updated the sentence for better clarity.

(Lines 144-145)

A longer past sequence may provide more information than a shorter one, but it could also contain unnecessary information for model training.

Section 3.2: What learning rates were used to train the models?

➔ The learning rate was set at 0.001 for the ADAM optimizer.

(Lines 221-223)

All models were trained with the MSE and BMSE loss functions and adaptive momentum optimizer (ADAM) with a learning rate of 0.001, widely adopted in deep learning regression models (Kingma and Ba, 2014).

Equations 2 and 3: Missing $1/n$.

➔ Thank you for pointing this out. We corrected the errors in Equations 1, 2, and 4. We also added balanced MSE (BMSE) to Equation 2.

(Line 155-159)

$$MSE = \frac{\sum_{i=1}^{N}(y_i-\hat{y}_i)^2}{N} \tag{1}$$

$$BMSE = \frac{\sum_{i=1}^{N} w(y_i)(y_i-\hat{y}_i)^2}{N}, w(y_i) = \begin{cases} 1, & y_i < 2 \\ 2, & 2 < y_i < 5 \\ 5, & 5 < y_i < 10 \\ 10, & 10 < y_i < 30 \\ 30, & y_i > 30 \end{cases}, \tag{2}$$

where y is the reference value, and $\hat{y}$ represents the predicted value and N is the number of all valid pixels within the radar area. Figure 2 shows the distribution of rainfall intensity and weights for BMSE.

(Line 276-277)

$$mean\ bias = \frac{\sum_{i=1}^{N}(y_i-\hat{y}_i)}{N} \tag{4}$$

where y is the reference value, and $\hat{y}$ represents the predicted value.

Table 5: Looking at these results, it seems that that there is no clear winner among these models. It really depends upon what we want to achieve (the performance metric used and the time-step in in future where we need the forecast). This is why I say that this project is a bit ambiguous. In fact, the main conclusion seems to be that there cannot be any specific guidelines for developing these models. I suspect the if the authors compute their goodness measures (CSI, MSE, etc.) separately for different region of the South Korea; they might find that different models perform better in different regions. These arguments apply to high rainfall case of Table 7 also.

➔ As you pointed out, the original manuscript's conclusion was ambiguous due to the absence of clear superiority among the compared schemes. In our revised version, we updated the key factors by dropping prediction design and maximum length and newly added the balanced loss function and ensemble approach. These updated factors convey a more significant comparison with conclusive discussion, as well as some enhanced results compared to the original design, as shown in the previous answers. Consequently, we updated the discussion about performance in Section 5.1.

5.1 Performance comparison and considerations of key factors

To address data imbalance and improve skill scores, various loss functions have been considered in previous research. Our comparison of two representative losses, MSE and BMSE, revealed that each has its strengths and weaknesses. The selection of an appropriate loss function should be informed by a comprehensive evaluation of QPN results. Optimal loss functions may vary depending on the specific objectives as it provides guidance to DL modeling. For instance, if the model's focus is on severe weather, BMSE can be weighted to emphasize high intensities. In cases where the area of precipitation over a certain threshold is of key interest, a modified CSI loss can be used (Ko et al., 2022). As a single metric cannot fully evaluate a model, combinations of different losses can also be explored. Alternatively, the ensemble approach analyzed in our study can leverage different loss functions to create a synergistic effect for QPN.

In this study, U-Net consistently outperformed ConvLSTM in various respects, both in long-term evaluation and in a single heavy rainfall event. This finding is in line with previous research (Ayzel et al., 2020; Ko et al., 2022; Han et al., 2023). Additionally, U-Net demonstrated more stability across different random seeds than ConvLSTM (Figure 8). Contrary to the widespread expectation that DL models powered by RNN would excel in time-series forecasting, it was found that a model relying solely on CNN can perform better. However, this does not imply that all models using RNN structures are inferior to full CNN *models. Considering the wide range of U-Net and ConvLSTM variants, there could be potential for RNN-powered models to exhibit superior results. Lastly, the input sequence length did not significantly impact the results compared to other factors in this study. Nevertheless, sequence length should still be carefully considered as DL-QPN relies significantly on past information. In other DL models and QPN designs, input sequence length may have a greater impact than it did in this study, therefore, we continue to regard this as a key factor in DL-QPN.*

Due to the inherent randomness and stochastic nature of deep learning, modeling and evaluation need to be carefully conducted, taking into account relevant factors. As demonstrated in Figure 8, results can vary for each run with a different random seed. Thus, stability should be a priority when developing a DL-QPN model, a point often overlooked in previous studies. By treating each run as an ensemble member, we can avoid unstable results under varying conditions of randomness.

Figures 5 and 6: Use a different color scheme to differentiate between the models? It is a bit difficult to differentiate between them.

- ➔ We believe that evaluation at every time step is not necessary. Therefore, we removed the time-series figures for 10-120 minutes. We have updated the figures for the time-series of heavy rainfall cases in the revised manuscript for better readability.

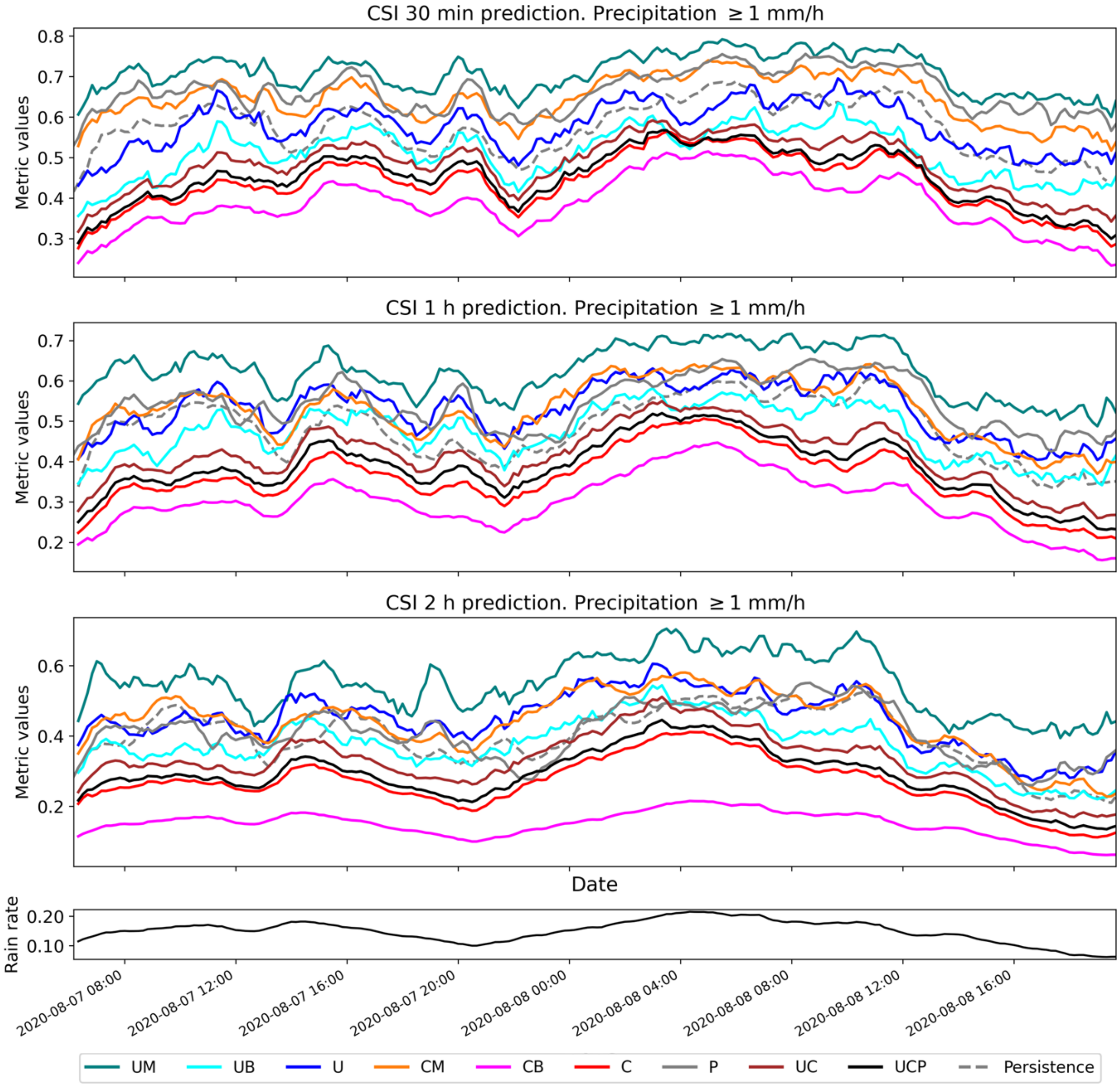


Figure 6. Comparison of CSI performance for the case of heavy rainfall over South Korea from 7th to 8th August 2020 with the 1 mm/h threshold. Refer to Table 2 for scheme names. The bottom black line represents the ratio of precipitation pixels > 1 mm/h for each radar scene.

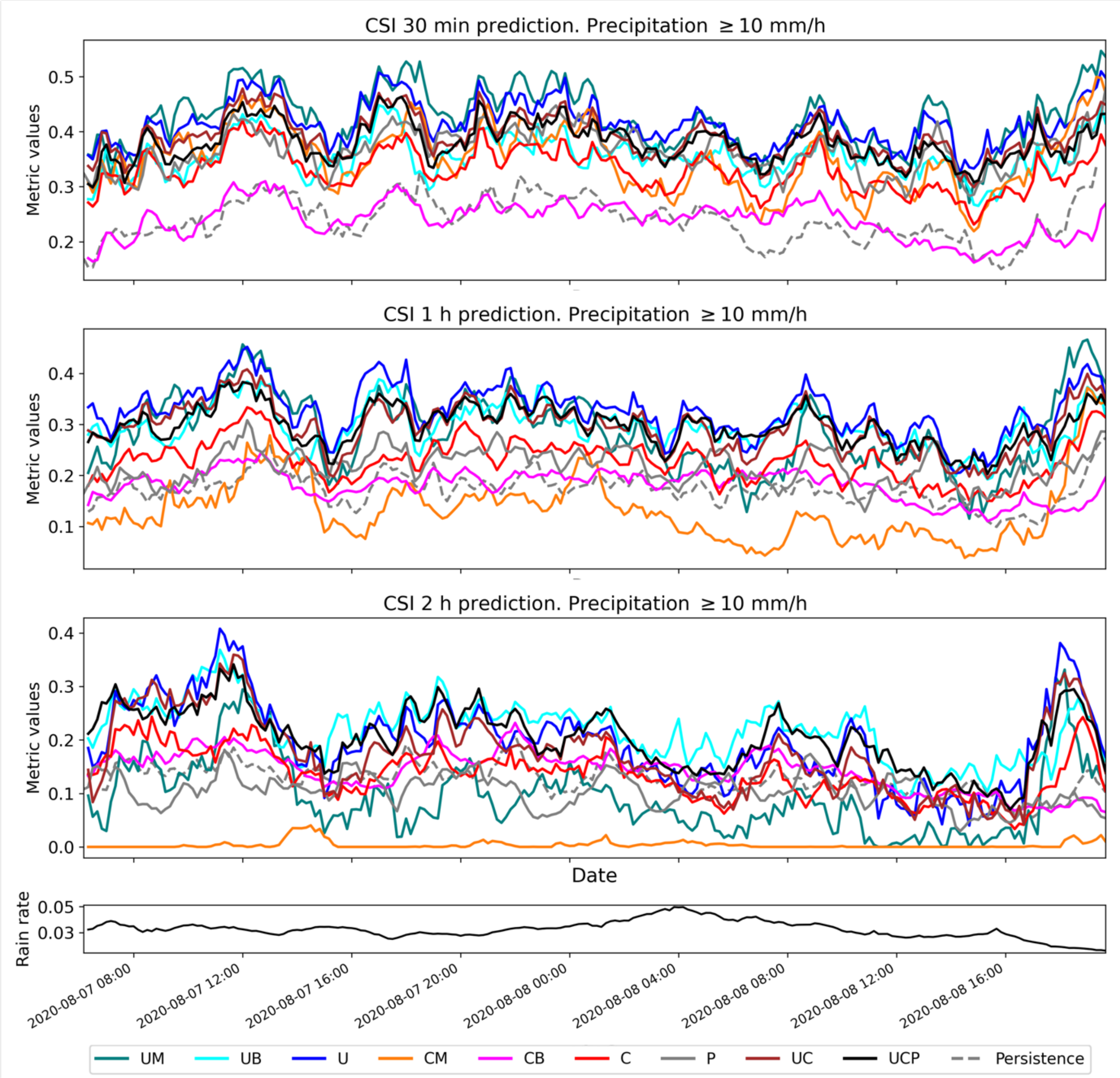


Figure 7. Comparison of CSI performance for the case of heavy rainfall over South Korea from 7th to 8th August 2020 with the 10 mm/h threshold. Refer to Table 2 for scheme names. The bottom black line represents the ratio of precipitation pixels > 10 mm/h for each radar scene.

Lines 294-296: These two sentences are not logically connected. Basically, the first part of the second sentence 'The two models produced identical 10-minute forecasts' can be removed.

Line 303: Language typo: gap was reduced with 5 mm/h threshold?

Line 306-307: This sentence is unclear. Rewrite.

Line 312: But even the CSI were not much better than persistence. So, does this sentence really make sense?

- ➔ In the revised version, these parts were removed as the result was updated with a new experimental design.

Lines 316-324; I am not sure the ConvLSTM has been designed properly. Basically, the sequence length of 12 may not be good enough for convLSTM.

- ➔ Based on your comment, we conducted experiments again for ConvLSTM, extending the input sequence to up to 3 hours, which is longer than the maximum lead time.

I think it still might be a good idea to include MAPLE results in Tables 5 and 7 even though these are based on different datasets. This would tell us at least something how a physics based model performs in comparison to the DL models.

- ➔ We agree with you that a comparison with a non-DL model would make this evaluation more meaningful. However, as MAPLE is based on a different data type with different quality control and masking, our concern is that it is hard to compare MAPLE (hybrid surface radar) with other DL models (radar CAPPI). Since MAPLE data was provided by the Korea Meteorological Administration, there was a limitation in changing the input data source. Hence, we replaced MAPLE with pySTEPS, a method widely used to demonstrate how non-DL models perform. By adopting pySTEPS, we can directly compare the results. The quantitative and qualitative comparisons with pySTEPS are provided in Tables 5 and 6, and Figures 6-8. As other results were already cited in the previous answers, we only cite Figure 8 here.

- ➔ Thank you for your comment. Similar to the previous comment, there was no quantitative evaluation using MAPLE. In lines 403-405, we discussed this based on the visual interpretation of Figures 11-14, which was still unclear. After replacing MAPLE with pySTEPS, we were able to calculate the CSI score, indicating how each model can predict the area over a given threshold. In the revised manuscript, the updated results now support that some DL models can achieve better performance than pySTEPS in terms of forecasting precipitation area.

3.2 Comparison with non-DL model

To compare the DL-QPN results with a non-DL model, we also included pySTEPS (Pulkkinen et al., 2019) in our comparison. PySTEPS is a Python implementation of the Short-Term Ensemble Prediction System (STEPS) proposed by Bowler et al. (2006). It has been widely used as a control non-DL model in previous studies (Ravuri et al., 2021; Choi and Kim, 2022; Han et al., 2023; Zhang et al., 2023). By calculating the mean wind vector using the input *radar sequence, pySTEPS simulates future radar sequences. To examine the impact of input sequence length, we also tested 1-3 h of input sequence in pySTEPS to predict a maximum of 2 h, the same as the other DL models. More detailed information and usage of pySTEPS can be found in its documentation (https://pysteps.github.io) and repository (https://github.com/pySTEPS/pysteps).*

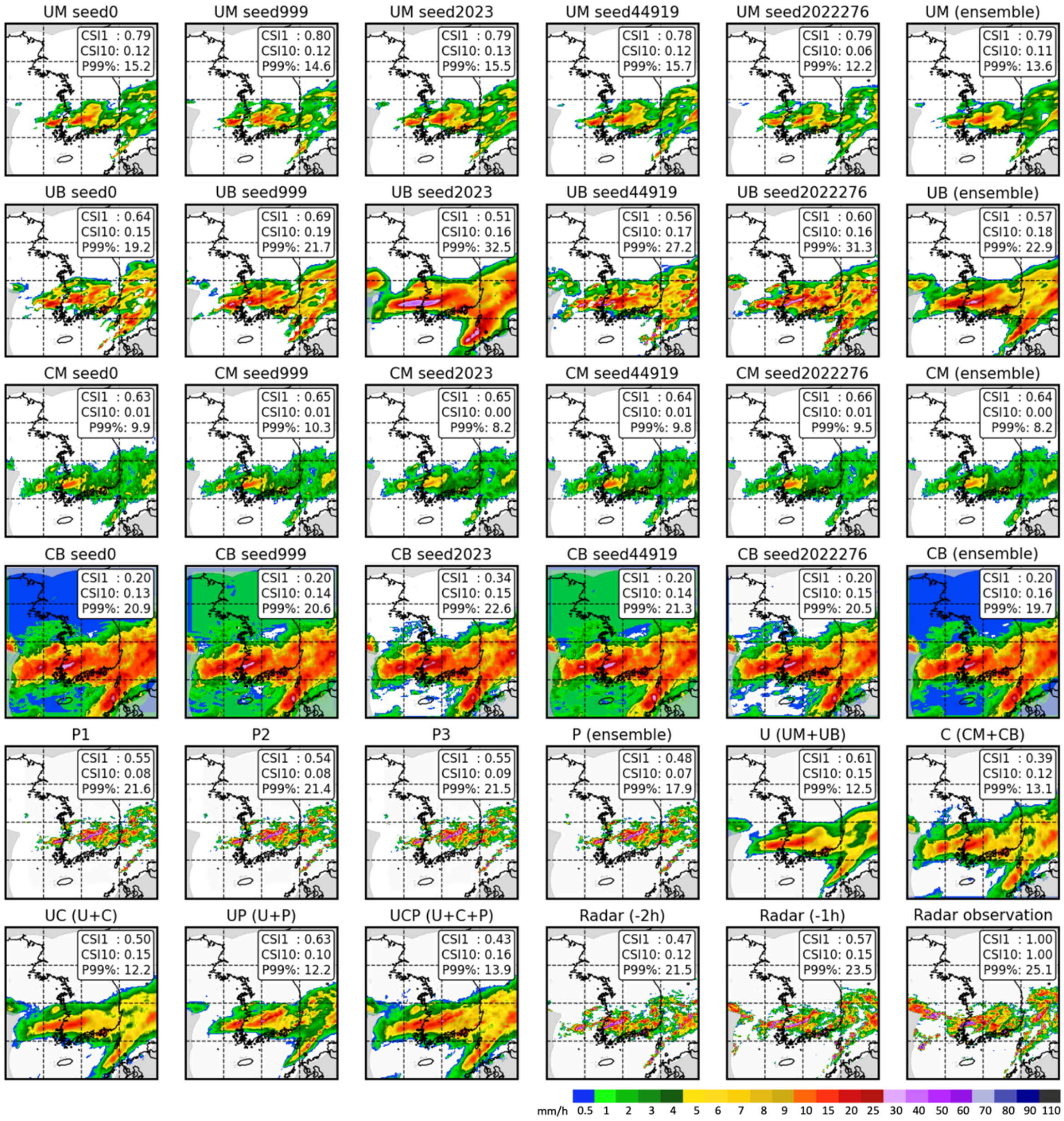


Figure 8. Comparison map for 05:00 on August 8, 2020, in KST with a 2h lead time. Refer to Table 2 for scheme names. Schemes with seed numbers are the averages of results for three scenes using 1h, 2h, and 3h input sequences. CSI1 and CSI10 indicate the CSI scores with thresholds of 1 and 10 mm/h, respectively.

Lines 403-405: This is not really supported from Figure 11-14.

➔ Thank you for your comment. Similar to previous comment, there was no quantitative evaluation using MAPLE. In lines 403-405, we discussed based on the visual interpretation of Figures 11-14, which is still not clear. After replacing MAPLE with pySTEPS, we can calculate CSI score which can indicate how each model can predict the area over the given threshold. In the revised manuscript, updated results now can support that some DL models can achieve better performance than pySTEPS in terms of forecasting precipitation area.

Lines 427-429: This is not really true. As I have mentioned earlier, ConvLSTM may perform better with larger sequence length.

➔ As in previous responses, we did recognize the need for testing with a longer input sequence. We extended the length up to 3 hours and found a performance improvement. Even though the performance slightly improved in terms of MSE or BMSE, U-Net still outperformed ConvLSTM in this time. Moreover, ConvLSTM exhibited higher variance depending on different random seeds and loss functions. The input length had little impact on both models.

➔ Based on a comment from RC1, we removed Section 4.2.3 to avoid a lengthy manuscript and to focus on other discussions. Therefore, these sentences were also removed from the revised manuscript.

Lines 447-448: This has not been mentioned earlier. Explain.

➔ This part was also removed to accommodate new results.