

---

# Self-Supervised Pre-Training for Precipitation Post-Processor

---

**Sojung An**<sup>\*†</sup>  
KIAPS  
Seoul, Republic of Korea  
sojungan@kiaps.org

**Junha Lee**<sup>\*</sup>  
KITECH  
Ansan, Republic of Korea  
junha@kitech.re.kr

**Jiyeon Jang**  
KIAPS  
Seoul, Republic of Korea  
jyjang@kiaps.org

**Inchae Na**  
KIAPS  
Seoul, Republic of Korea  
icna@kiaps.org

**Wooyeon Park**  
KIAPS  
Seoul, Republic of Korea  
wooyeon@kiaps.org

**Sujeong You**  
KITECH  
Ansan, Republic of Korea  
sjyou21@kitech.re.kr

## Abstract

Obtaining a sufficient forecast lead time for local precipitation is essential in preventing hazardous weather events. Global warming-induced climate change increases the challenge of accurately predicting severe precipitation events, such as heavy rainfall. In this paper, we propose a deep learning-based precipitation post-processor for numerical weather prediction (NWP) models. The precipitation post-processor consists of (i) employing self-supervised pre-training, where the parameters of the encoder are pre-trained on the reconstruction of the masked variables of the atmospheric physics domain; and (ii) conducting transfer learning on precipitation segmentation tasks (the target domain) from the pre-trained encoder. In addition, we introduced a heuristic labeling approach to effectively train class-imbalanced datasets. Our experiments on precipitation correction for regional NWP show that the proposed method outperforms other approaches.

## 1 Introduction

In modern society, precipitation forecasting plays a vital role in the response to and prevention of social and economic damage. Deep learning has been rapidly utilized in precipitation forecasting, such as for simulating echo movements and predicting typhoon trajectories based on observational data [12, 13, 10, 1]. However, exclusively relying on observational data fails to capture the fundamental physical and dynamic mechanisms in the real world. This results in an exponential increase in error as the forecast lead time increases [3]. In addition, climate change increases the uncertainty of predicting extreme events such as torrential rains [9]. A limited forecast lead time makes it difficult to prepare for extreme weather events in advance.

Recent research has been actively conducted to enhance forecast lead times via the post-processing of numerical weather prediction (NWP) model data [8, 6, 11, 16]. Espenholt et al. [3] proposed Metnet2, a 12-h probabilistic forecasting model. They designed a hybrid model consisting of a forecasting model based on observational data and post-processing NWP model data. While Metnet2 displayed promising results, simulating heavy rainfall remains a challenging task.

Therefore, we designed a self-supervised pre-training process that considers the physical and dynamic processes among atmospheric variables to improve the relative bias and reliability of predicting

---

<sup>\*</sup>These authors contributed equally to this work.

<sup>†</sup>Corresponding author

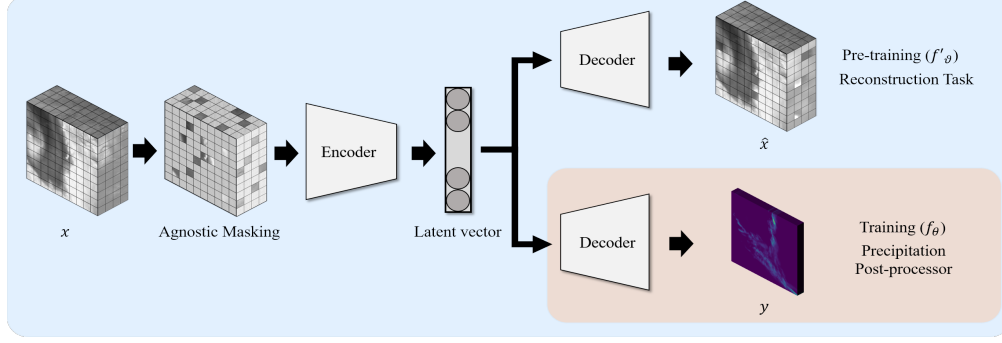


Figure 1: Process of learning the precipitation post-processor. The model consists of two main phases: 1) pre-training the encoder and decoder using a reconstruction task after masking the inputs and 2) training the decoder for precipitation prediction using the trained encoder. During the main training, the latent vector learned during pre-training is used as an encoder with fixed weights.

heavy rain. To this end, we randomly masked three-dimensional (3D) images [5] and trained an encoder-decoder to reconstruct variables based on dynamic sparse kernels (DSKs) [14]. Subsequently, we trained a decoder for probability-based precipitation correction based on a pre-trained encoder. The pre-training process helps in conducting a correlation analysis based on the NWP model variables and provides additional data-augmentation effects. Finally, we propose a continuous labeling method for learning class-imbalanced datasets. This method facilitates a continuous probability distribution based on the precipitation density.

## 2 Method

This section describes the proposed approach. We first present a self-supervised pre-training procedure that aims to define the reconstruction task. Subsequently, we describe the target process for the precipitation post-processing.

### 2.1 Problem formulation

Given a set of ground-truth pairs  $\mathbf{G} = (\{x_i\}_{i=1}^n, \{y_j\}_{j=1}^m)$  with  $x \in \mathbb{R}^{n \times h \times w}$ ,  $y \in \mathbb{R}^{m \times h \times w}$ , and  $z \in \mathbb{R}^d$ , let  $x = \{x_1, \dots, x_n\}^T$  represent the input, and  $y = \{y_1, \dots, y_m\}^T$  represent the output. Our objective is to derive a precipitation segmentation function  $f : \mathbb{R}^d \rightarrow \mathbb{R}^c$ , assuming  $z \in \mathbb{R}^d$  as a latent space with distribution  $p(z)$  defined by the pre-training process. The task involves reconstructing  $f'_\vartheta$  from the NWP data to predict the target object function  $f_\theta$ . The probabilistic encoder-decoder of the proposed method is defined as follows:

$$\mathcal{L}(\theta, \vartheta; x) = -\frac{1}{2} \mathbb{E}_{q_\vartheta(z|x)} [\|x - \hat{x}\|_2^2] + \text{KL}(q_\vartheta(z|x) | p^*(y)), \quad (1)$$

where  $\hat{x}$  denotes the reconstructed output from  $x$  as generated by decoder,  $q_\vartheta$  represents the pre-training process distribution, and  $p^*$  is the class probability distribution, detailed in Section 2.2. The loss function combines the mean squared error for minimizing pre-training loss and the Kullback-Leibler (KL) divergence term to quantify the difference between the learned latent variable distribution  $q_\vartheta(z|x)$  and the class probability distribution  $p^*(y)$ .

### 2.2 Continuous labeling

In multi-class classification, instances  $x$  are classified into one of the  $y$  labels based on the ranges of precipitation intensity. Let  $p^*(y) = [\mathbb{P}(y|z)]$  be the class probability distribution and  $\mathcal{L}$  be the cross-entropy loss. For the predictive function  $f(x)$  and the label  $y$ , the cross-entropy loss  $\mathcal{L}(f(x), y)$  is computed as follows:

$$\mathcal{L}(f(x), y) = -\sum_{j=1}^m w_j \cdot p^*(y_j) \log(f(x)_j), \quad (2)$$

where  $w_j$  denotes the weight set according to each class label  $y_j$ , and  $p^*(\cdot)$  denotes the continuous labeling. The minimization leads to the maximum likelihood estimate of the classifier parameters. Minimization yields a maximum likelihood estimate of the classifier parameters. Algorithm 1 presents the method for smoothing the probability values based on the density of the label range. This method differs from that of one-hot labeling, where the probability takes a value of one. In the smoothing method, the sum of the probability values per class is fixed at one.

---

**Algorithm 1:** Continuous labeling of the density of the rainfall range

---

**Input:** QPE  $y = [0, 100]$ ; rainfall threshold set  $\gamma = \{r_0, \dots, r_{m-2}\}$ , ( $m \geq 2$ )

**Output:** Probability label  $\hat{Y}$

```

1 Initialize the number of rainfall thresholds  $\gamma$ 
2  $p(y_j) = \begin{cases} 1, & \text{if } r_{j-1} \leq y < r_j \quad /* \text{ Set the probability of label } k \text{ to } 1. */ \\ 0, & \text{otherwise} \quad /* \text{ Set the probability of label } i \text{ to } 0 \text{ if outside the range. */} \end{cases}$ 
3 while  $j$  do
4   switch  $p(y_j)$  do
5     if  $\{x \leq r_{j-1}\}$   $p^*(y_j) \rightarrow 0$ 
6     else if  $\{r_{j-1} < y_j \leq r_j\}$   $p^*(y_j) \rightarrow \frac{r_{j+1}-y_j}{r_{j+1}-r_j}$ 
7     else if  $\{r_j < y_j \leq r_{j+1}\}$   $p^*(y_j) \rightarrow 1 - \frac{y_j-r_j}{r_j-r_{j-1}}$ 
8     else if  $\{j = m - 2\}$   $p^*(y) \rightarrow 1$ 
9     else  $\implies p^*(y_j) \rightarrow 0$ 
10  end
11 end
12 return  $p^*(y)$ 

```

---

Given a rainfall  $y_c$  that lies between two thresholds  $r_j$  and  $r_{j+1}$ , assume that  $y_c$  is closer to  $r_{j+1}$ . The probability values for each threshold are sets  $p(y_{c=j}) = 1$  and  $p(y_{c \neq j}) = 0$ . In this context,  $p(\cdot)$  denotes the probability value based on the original labeling. According to Algorithm 1, both  $p^*(y_j)$  and  $p^*(y_{j+1})$  are non-zero,  $p^*(y_c)$  has a probability value of  $p^*(y_j) < p^*(y_{j+1})$ , and  $p^*(y_j) + p^*(y_{j+1}) = 1$ . Setting the probability value using one-hot labeling can be a limitation in learning the uncertainty of the NWP. By continuously smoothing the probability value of a label, the method can help reduce uncertainty about the likelihood of being a different label.

### 2.3 Training model

This section presents our post-processing method that perturbs an entire patch for the reconstruction task. This method uses InternImage [14] as an encoder and UPerNet [15] as a decoder.

**Patch embedding.** We tokenize the input data into nonoverlapping spatial-temporal patches [4]. Each 3D patch has dimensions  $\mathcal{M} \in \mathbb{R}^{t \times p \times p}$ , where  $t$  and  $p$  denote token size of time and height/width respectively. This patching approach results in  $x \in \mathbb{R}^{\frac{n}{t} \times \frac{h}{p} \times \frac{w}{p}}$  tokens. We then flatten the data and transform the tokens into  $x \in \mathbb{R}^{\frac{nhw}{4p^2} \times C}$  using a projection process. The data is added spatio-temporal positional encodings of the same size.

**Masking.** Based on reference [5], we use a structure-agnostic sample strategy to randomly mask the patches without using replacements from the set of embedded patches. For a pre-training model in a reconstruction task, the optimal masking ratio is related to the amount of redundant information in the data [2, 7]. Numerical forecasting models have a similar information redundancy, as each weather variable is at the same point in time. However, each variable has its own physical information. Based on this, we use empirical results to determine the masking ratio. The masking ratios of the pre-training and training are set to 90% and 25%, respectively. The masked patches underwent layer normalization and are restored to their original input dimensions of  $x' \in \mathbb{R}^{n \times h \times w}$ .

**Encoder and decoder.** We utilize DSK layers in the InternImage [14] encoder for adaptive spatial aggregation. InternImage is a backbone model that employs various techniques, in addition to attention, in the receptive field required for downstream tasks. InternImage applies each encoder layer  $e_i \subset e$  to a hierarchy of hidden states and the weights of each layer are stacked. We use the

Table 1: Summary of results for our tasks.  $\mathcal{I}$ ,  $p^*$ , and  $\vartheta$  denote InternImage, label smoothing, and transfer learning, respectively. The tables compare the results of the models with thresholds of 0.1 and 10 mm.

Method	0.1 mm $\uparrow$				10 mm $\uparrow$			
	CSI	F1	Precision	Recall	CSI	F1	Precision	Recall
RDAPS	0.296	0.456	0.405	0.522	0.062	0.117	0.077	<b>0.238</b>
Metnet	0.276	0.433	0.427	0.439	0.007	0.015	0.015	0.274
Metnet+ $p^*$	0.271	0.427	0.390	0.472	0.057	0.107	0.002	0.275
$\mathcal{I}+p^*$	0.316	0.488	0.488	0.502	0.017	0.020	0.020	0.132
$\mathcal{I}+\vartheta$	0.329	0.443	0.443	<b>0.523</b>	0.060	0.076	0.075	0.222
$\mathcal{I}+p^*+\vartheta$ (ours)	<b>0.347</b>	<b>0.515</b>	<b>0.620</b>	0.481	<b>0.093</b>	<b>0.169</b>	<b>0.130</b>	0.227

UpperNet decoder to enable effective segmentation while preserving object boundaries and details from the training data.

### 3 Experiments

We classified rainfall up to  $[0, 0.1)$  as no rain,  $[0.1, 10)$  as rain, and above 10 as heavy rain. Figure 2 (a) shows the proportions of each label. The proportion of heavy rain within the training dataset was approximately 0.75%. This indicates a significant data imbalance. The class imbalance issue frequently observed in precipitation measurement data is a common challenge in precipitation forecasting. Therefore, we investigated how the proposed method addresses class imbalance issues during the model training process. We compared and evaluated our post-processor and heuristic label approaches with the state-of-the-art precipitation forecasting model called Metnet.

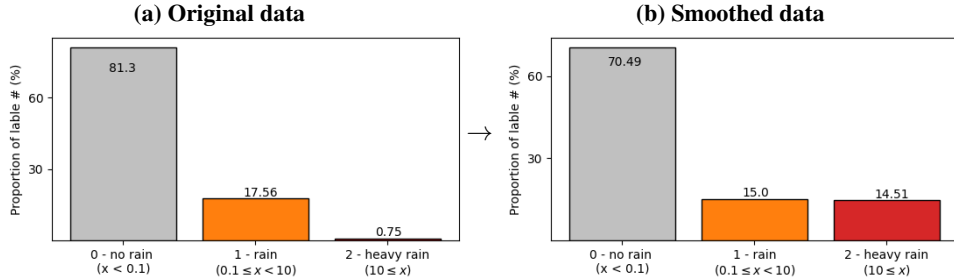


Figure 2: Visualization of the proportion of labels in the training dataset: (a) is the original data and (b) shows the proportion for pixels with a non-zero probability smoothed using the method proposed in Section 2.2.

Our main results are summarized in Table 1. For the pre-trained model, we used a mask ratio of 90%, a learning rate of  $1.6e-3$ , and approximately 150 k iterations. For the training model, we used a mask ratio of 25%, a learning rate of  $1e-4$ , and approximately 35 k iterations. Training the model using continuous labeling had a noticeable impact on the performance improvement. We observed a significant enhancement in the accuracy of heavy rain when employing continuous labeling in Metnet. We aimed to address (i) solving the imbalanced label problem and (ii) learning the weights between the variables in the self-supervised learning.

Figure 3 shows the reconstruction of 16 masked variables. Using the pre-trained model to understand the physical flow based on changes in each variable and across vertical levels is crucial for precipitation prediction. Despite masking an overload of information in each variable, the NWP model accurately predicted prominent patterns across all variables. In particular, we observed a resemblance in the predicted patterns for the ‘rain’ variable despite the masked pixels carrying limited information for this instance and inherent nonlinearity of the variable.

Figure 4 shows that above 10 mm of precipitation covers much of the Korean Peninsula. A deep learning-based post-processing model captures above 10 mm of rainfall, while NWP models (RDAPS) underestimate this case. In Figure 4 (c), when trained with Algorithm 1, Metnet exhibits errors in the rainfall location but demonstrated accurate predictions for 10 mm of rainfall. In Figure 4 (d), for the

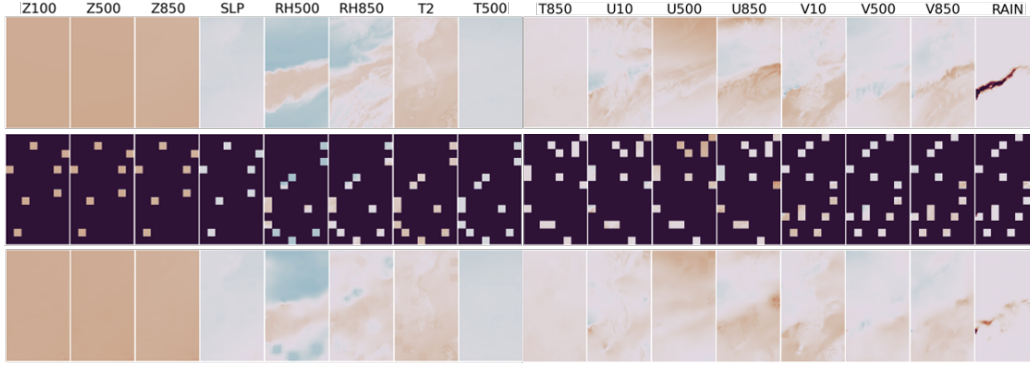


Figure 3: Variable reconstruction results using the pre-trained model on data from August 10, 2022 at 00 UTC. The first row visualizes the normalized variables. The second row visualizes the variables with 90% of the pixels masked. The third row shows the results of reconstructing the masked pixels. For the visualization, the masked values were set to -100, and a range of (-10, 10) was used. The number beside the variable indicates the vertical level.

proposed model, we observed predictions for rainfall location and 10 mm of rainfall, but these tended to be overestimated. Based on the results, combining  $p(y)$  and  $p^*(y)$  is expected to further enhance the accuracy. The limitation lies in the fact that the input data relies on NWP predictions. As a result, the prediction model learns with a certain margin of error and predicts in a manner similar to RDAPS in some cases.

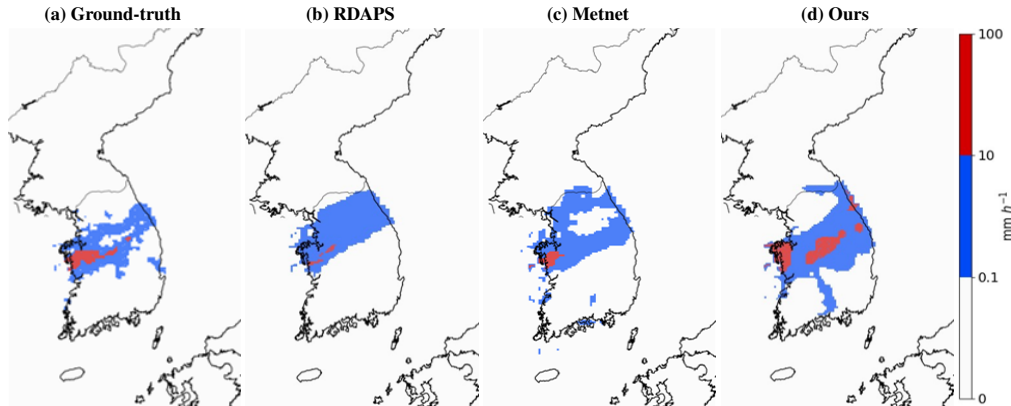


Figure 4: Qualitative comparison between models trained on data from August 10 2022 at 00 UTC. Each result represents a cumulative result over a 1-hour period. Owing to the influence of a stagnant front, rain fell in most parts of Korea; the average rainfall is 100~200 mm per day, and the maximum exceeds 300 mm.

## 4 Conclusion

This study proposes an approach for precipitation post-processing based on transfer learning for target domain adaptation. Our main contributions involve (1) applying transfer learning to the atmosphere system by conducting pre-training on the reconstruction domain and integrating the parameters in the segmentation domain and 2) employing stochastic softening one-hot labeling to overcome biased learning from unbalanced data. Our experiments on domain adaptation using the two-step training strategy show that the proposed method helps understand the correlation among atmospheric variables. This yields a better transfer learning performance from the precipitation post-processor.

## 5 Acknowledgments

This work was carried out through the R&D project “Development of a Next-Generation Operational System by the Korea Institute of Atmospheric Prediction Systems (KIAPS)”, funded by the Korea Meteorological Administration (KMA2020-02213).

## References

- [1] Sojung An. Nowcast-to-forecast: Token-based multiple remote sensing data fusion for precipitation forecast. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023.
- [2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [3] Lasse Espeholt, Shreya Agrawal, Casper Sønderby, Manoj Kumar, Jonathan Heek, Carla Bromberg, Cenk Gazen, Rob Carver, Marcin Andrychowicz, Jason Hickey, et al. Deep learning for twelve hour precipitation forecasts. *Nature communications*, 13(1):1–10, 2022.
- [4] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835, 2021.
- [5] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. *Advances in neural information processing systems*, 35:35946–35958, 2022.
- [6] Mohammadvaghef Ghazvinian, Yu Zhang, Dong-Jun Seo, Minxue He, and Nelun Fernando. A novel hybrid artificial neural network-parametric scheme for postprocessing medium-range precipitation forecasts. *Advances in Water Resources*, 151:103907, 2021.
- [7] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.
- [8] Taehyeon Kim, Namgyu Ho, Donggyu Kim, and Se-Young Yun. Benchmark dataset for precipitation forecasting by post-processing the numerical weather prediction. *arXiv preprint arXiv:2206.15241*, 2022.
- [9] Katrin M Nissen and Uwe Ulbrich. Increasing frequencies and changing characteristics of heavy precipitation events threatening infrastructure in europe under climate change. *Natural Hazards and Earth System Sciences*, 17(7):1177–1190, 2017.
- [10] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. *Advances in neural information processing systems*, 32, 2019.
- [11] Adrian Rojas-Campos, Martin Wittenbrink, Pascal Nieters, Erik J Schaffernicht, Jan D Keller, and Gordon Pipa. Postprocessing of nwp precipitation forecasts using deep learning. *Weather and Forecasting*, 38(3):487–497, 2023.
- [12] Xingjian Shi, Zhoung Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.
- [13] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. *Advances in neural information processing systems*, 30, 2017.
- [14] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14408–14419, 2023.

- [15] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.
- [16] Yuhang Zhang and Aizhong Ye. Machine learning for precipitation forecasts postprocessing: Multimodel comparison and experimental investigation. *Journal of Hydrometeorology*, 22(11): 3065–3085, 2021.