Nowcasting of Extreme Precipitation Using Deep Generative Models

Bi, Haoran ; Kyryliuk, Maksym ; Wang, Zhiyi ; Meo, Cristian; Wang, Yanbo; Imhoff, Ruben; Uijlenhoet, Remko; Dauwels, Justin

DOI
10.1109/ICASSP49357.2023.10094988

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10094988

NOWCASTING OF EXTREME PRECIPITATION USING DEEP GENERATIVE MODELS

Haoran Bi[1], Maksym Kyryliuk[1], Zhiyi Wang[1], Cristian Meo[1], Yanbo Wang[1], Ruben Imhoff[2], Remko Uijlenhoet[1], Justin Dauwels[1]

[1]Delft University of Technology, Netherlands
[2]Deltares, Netherlands

ABSTRACT

Nowcasting is an observation-based method that uses the current state of the atmosphere to forecast future weather conditions over several hours. Recent studies have shown the promising potential of using deep learning models for precipitation nowcasting. In this paper, novel deep generative models are proposed for precipitation nowcasting. These models are equipped with extreme-value losses to more reliably predict extreme precipitation events. The proposed deep generative model contains a Vector Quantization Generative Adversarial Network and a Transformer ("VQGAN + Transformer"). For enhanced modeling and forecasting of extreme events, Extreme Value Loss (EVL) is incorporated in the autoregressive Transformer. The numerical results show that the proposed model achieves comparable performance with the state-of-the-art conventional nowcasting method PySTEPS for predicting nominal values. By incorporating an EVL, the proposed model yields more accurate nowcasting of extreme precipitation.

1. INTRODUCTION

1.1. Existing Nowcasting Methods

Extreme precipitation often causes serious hazards such as flooding and landslides, which pose threats to human lives and cause substantial economic loss. In order to give better early warnings of such hazards, nowcasting systems have been widely used to forecast the future weather condition in the short term (typically less than 6 hours), which is a timeframe in which numerical weather prediction (NWP) models have limited use [1]. Even though the nowcasting system can only predict weather conditions for the following few hours, accurate and reliable nowcasting results are essential for the early warning of serious extreme-precipitation-related hazards [2]. In the field of precipitation nowcasting, weather conditions are usually represented by the radar precipitation fields produced by weather radars [1]. The inputs of a nowcasting system are precipitation fields of the previous timestamps and the outputs are forecasting of future radar precipitation fields. Conventional radar-extrapolation methods are the basis of most operational nowcasting systems nowadays.

Researchers have also explored the possibility of using deep learning models for nowcasting tasks. Similar to the conventional nowcasting methods, most deep learning models also use radar precipitation fields to represent weather conditions and try to extrapolate the future precipitation field. The first deep learning precipitation nowcasting model was proposed by Shi [3] in 2015, called ConvLSTM. This model considers nowcasting as a video prediction task and applies convolution operation in LSTM to capture spatial and temporal features at the same time. Another group of researchers considered it an image transformation task and designed the deep neural network model inspired by the U-Net structure (e.g., [4, 5]). Despite their success in many other tasks, deep generative models such as GANs and VAEs have not been widely applied for nowcasting tasks. However, as shown in [6] and [7], the deep generative models have great potential for producing skillful nowcasting results.

Compared with traditional radar-extrapolation methods, deep learning models are purely data-driven, flexible, and require no explicit physics constraints [6]. However, the precipitation nowcasting tasks are challenging and different from other well-developed deep learning tasks. The precipitation maps predicted by deep learning models tend to be blurred, moreover, these models tend not to capture extreme precipitation patterns [1]. In this paper, we explore using deep generative models to avoid blurry generation and incorporating extreme value theory to capture extreme patterns. The result indicates that our method is promising for overcoming these problems.

1.2. Dataset

We consider in this paper nowcasting for the Netherlands. Specifically, we analyze radar data from the Royal Netherlands Meteorological Institute (KNMI). The selected dataset contains the radar reflectivity data from 2008 to 2021, with a spatial resolution of 1 km and temporal resolution of 5 minutes. The rainfall rate can then be estimated from the radar reflectivity using a Z-R transformation [8]. A river catchment-level analysis is conducted for this dataset in order to assess the viability of the model on the scale of real-life applications.

Catchments mark the boundaries of the land surface area where all rainfall (eventually) ends up in the same river system. High rainfall amounts in the catchment can lead to flooding and nowcasting results of the catchment area are crucial for flood early warning. In this work, 12 Dutch lowland catchments were analyzed in a similar way to [9]. The locations of the catchments are shown in Figure 1. For the analysis, we looped through every possible event starting time, so all 3-hour events between 2008-2014 are examined. The average rainfall accumulation within catchment areas during this three-hour time window is calculated and used as the main indicator of rain intensity level for the catchment area. Table 1 summarizes the analysis result (R is the catchment average precipitation accumulation over 3 hours). From the analysis, we can conclude that the distribution of precipitation intensity is highly imbalanced, with more than 90% of the catchment-averaged accumulation smaller than 1mm, while the highest value can be larger than 25mm.

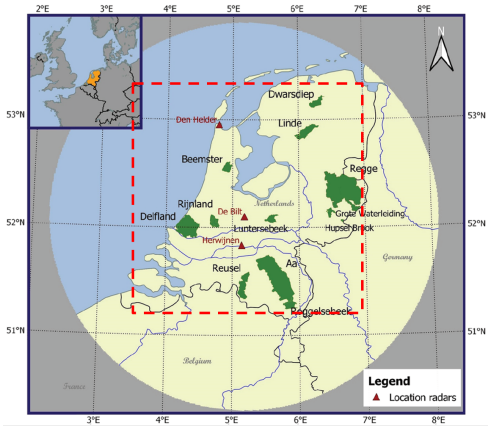

Fig. 1. Map of the Netherlands with the catchment areas marked in green and the study area marked by the red box.

Accumulation R	Occurence	Percentage
$X \leq 1$mm	1,245,834	91.1%
1mm $< X \leq$ 3mm	79,907	5.80%
3mm $< X \leq$ 5mm	24,484	1.77%
5mm $< X \leq$ 7mm	9,192	0.67%
7mm $< X \leq$ 9mm	4,141	0.30%
$X >$ 9mm	3,856	0.28%

Table 1. Summary of catchment-level data analysis.

1.3. Problem Formulation

The model is expected to fulfill two goals: output skillful precipitation nowcasting results for the Netherlands and reliably forecast extreme rainfall events happening within the catchment areas.

This work aims at nowcasting with a maximum lead time of 3 hours and time intervals of 30 minutes, so the forecasting output of the model contains 6 precipitation fields (T+30, T+60, T+90, T+120, T+150, T+180 minutes). Most deep learning nowcasting models use radar data of the last 30 to 90 minutes as their input (e.g., [3, 5, 6]). For our model, precipitation fields of the previous 60 minutes are used as input (T-60, T-30, T minutes). The KNMI dataset provides radar maps with a shape of 765×700, temporal resolution of 5 minutes and spatial resolution of 1 km. Since we focus on catchments areas, we consider a 256×256 study area (shown in Figure 1) that covers most of the national land area and all catchment areas.

Typically, extreme rainfall is defined by the distribution of yearly maximum rainfall. However, since we only consider 14 years of data, the number of annual maxima is too small for training and testing models. Therefore, we relax this definition and consider the precipitation in a catchment to be extreme when the 3-hour average precipitation in that catchment belongs to the top 1% for the period of 2008 to 2021.

2. METHODOLOGY

2.1. Model Architechture

The proposed "VQGAN + Transformer" follows a similar general structure as "NÜWA", which is a two-stage deep generative model

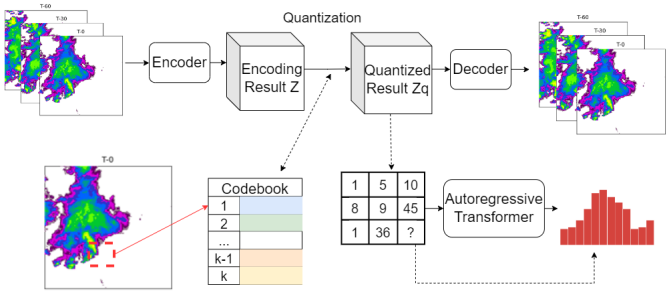

Fig. 2. The overall structure of the proposed model.

proposed in [10]. Such two-stage deep generative models have achieved state-of-the-art performance in multiple visual synthesis tasks, such as video prediction [10] and image synthesis [11]. The overall structure of the proposed model is shown in Figure 2. In the first stage, the VQGAN [11] learns a codebook, and each code (or combination of codes) is a representation of certain patterns of the radar precipitation field. The data can then be compressed and represented by a sequence of indices, whose composition is modeled subsequently by the autoregressive Transformer [12]. When forecasting, a sequence of condition indices is sent to the Transformer, which can then generate probability distributions of prediction tokens in an autoregressive way.

2.1.1. VQGAN

The first stage model VQGAN contains two main components: a Vector Quantized Variational Autoencoder (VQVAE) as generator and a patch-based discriminator. When training, the VQVAE can generate a reconstruction of the precipitation field while the discriminator is trying to discriminate the reconstruction from the original precipitation fields. The VQVAE's loss function is given by:

$$\mathcal{L}_{VQ} = \|I - \hat{I}\|_2^2 + \|\text{sg}[E(I)] - B[z]\|_2^2 + \|E(I) - \text{sg}[B[z]]\|_2^2, \quad (1)$$

where E and B are the encoder and codebook respectively. The first term is the reconstruction loss between the input I and the reconstructed input $\hat{I}$. The second term is used to update the code-book and is called the "commitment los". This loss term penalizes the difference between the encoder output and codebook. The gradients of the parameters of the encoder are not calculated (stop gradient, or sg) and only the codebook will be updated. The third term is the same as the second but with a stop gradient operation on the codebook, so only the encoder will be updated to get encoder output close to the vectors in the codebook.

For adversarial training, the discriminator D tries to maximize $\mathcal{L}_D = \log D(I) + \log(1 - D(\hat{I}))$, while the generator G tries to minimize $\mathcal{L}_G = \mathcal{L}_{VQ} + \log(1 - D(\hat{I}))$. The overall optimization problem for this first stage model is as follows:

$$\begin{aligned} \arg\min_{E,G,B} \max_{D} E_{x\sim p(x)} [\mathcal{L}_{\text{VQ}}(E, G, B) \\ + \lambda \mathcal{L}_{\text{GAN}}(\{E, G, B\}, D)], \end{aligned} \quad (2)$$

where $\mathcal{L}_{\text{GAN}}$ is the sum of $\mathcal{L}_G$ and $\mathcal{L}_D$, and λ is an adaptive weight:

$$\lambda = \frac{\|\nabla_{G_L} [\mathcal{L}_{\text{VQ}}]\|}{\|\nabla_{G_L} [\mathcal{L}_{\text{GAN}}]\| + \delta}, \quad (3)$$

where $\nabla_{G_L}[L]$ is the loss function's gradient concerning the last layer of the generator, and δ is a small number for stability.

2.1.2. Autoregressive Transformer

The autoregressive Transformer is a variant of the original Transformer [12] and has been widely applied in generation tasks (e.g., [13, 14]). For an autoregressive Transformer, a sequence of previous tokens is fed into multiple attention blocks, which can output a probability distribution for the next token. Instead of the full attention used for the original transformer, a sparse attention layer called 3DNA [10] is used for the precipitation nowcasting. The 3DNA takes advantage of the 3D shape of our data (sequence of 2D precipitation fields) and allows more efficient training. For a token tensor, each token of location (i, j, k) only pays attention to tokens within the cube of size (e^h, e^w, e^t) around this location. Weight parameters can then be applied for the neighbourhood tokens to calculate the query, key, and value (Q, K, and V). The output can be expressed as:

$$y_{ijk} = \text{softmax}\left(\frac{\left(Q^{(i,j,k)}\right) K^{(i,j,k)T}}{\sqrt{d^{in}}}\right) V^{(i,j,k)}, \tag{4}$$

where d^{in} is the size of token set. During training, the prediction output is first encoded into tokens and used as the input. Since the prediction needs to be causal, the tokens are right-shifted, and all tokens behind the current token are masked. Next, the tokens are fed into L layers of the 3DNA block, where the sequence computes its cross-attention to the condition tokens and self-attention to the output sequence of the previous layer:

$$\begin{aligned} Y^{\ell}_{ijk} = &\text{3DNA}\left(Y^{\ell-1}_{<i,<j,<k}, Y^{\ell-1}_{<i,<j,<k}\right) \\ &+ \text{3DNA}\left(Y^{\ell-1}_{<i,<j,<k}, C\right), \end{aligned} \tag{5}$$

where Y^{ℓ} is the output of the ℓ^{th} layer, and C is the latent space representation of the observed radar precipitation field. The final output is a set of probabilities for different tokens in the codebook. Therefore, it can be viewed as a multi-class classification task, and the cross-entropy is a suitable loss for training this model.

2.2. Extreme Value Loss (EVL)

The cross entropy is the standard loss function used for training the Transformer. However, the imbalance shown in the data analysis also leads to imbalance in the tokens and thus poor classification performance. A classic solution to the imbalance problem is to assign weights to the different classes, where large weights are assigned to the minority classes. In our case, a weighted cross entropy can be applied, with weights inversely proportional to the occurrence of the tokens. This technique is usually applied to solve small imbalance problems, whereas our dataset can be highly imbalanced (potentially 1:10,000 between minority and majority tokens). To further improve the extreme event modelling ability, we also explore the extreme value loss (EVL) to train the Transformer.

The EVL has been proposed in [15] for modeling extreme events in time series. The loss function is based on extreme value theory (EVT) to model the tail of a distribution. Specifically, from EVT [16, 17], the tail distribution of real-world data y can be modeled as:

$$1 - F(y) \approx (1 - F(\xi)) \left[1 - H\left(\frac{y - \xi}{f(\xi)}\right)\right], y > \xi, \tag{6}$$

where ξ is the threshold defining extreme values, F is the probability distribution, f is a scale function, and H is the Generalized Pareto Distribution (GPD) expressed as $H(x) = 1 - (1 - \frac{x}{\gamma})^{\gamma}$.

Viewing the detection of extreme events as a binary classification task, the cross-entropy loss can then be used to train this binary detector. In addition, the term $\frac{y-\xi}{f(\xi)}$ can be approximated by an extreme indicator u_t, which indicates the probability of the current predicted value being an extreme value. The tail distribution can be approximated from equation (6) and used as the weights for the cross-entropy loss. The loss can be expressed as [15]:

$$\begin{aligned} &\text{EVL}\left(u_t\right) = -\left(1 - P\left(v_t = 1\right)\right)\left[1 - H(u_t)\right] v_t \log\left(u_t\right) \\ &- \left(1 - P\left(v_t = 0\right)\right)\left[1 - H(1 - u_t)\right]\left(1 - v_t\right) \log\left(1 - u_t\right) \\ &= -\beta_0 \left[1 - \frac{u_t}{\gamma}\right]^{\gamma} v_t \log\left(u_t\right) \\ &- \beta_1 \left[1 - \frac{1 - u_t}{\gamma}\right]^{\gamma} \left(1 - v_t\right) \log\left(1 - u_t\right), \end{aligned} \tag{7}$$

where $v_t \in \{0, 1\}$ is the ground truth extreme indicator, and γ is a hyper-parameter. Intuitively, for the weights, the terms β_0 and β_1 are proportions of non-extreme and extreme tokens estimated from the training set and handle the imbalance between extreme and non-extreme tokens. The terms $\left[1 - \frac{u_t}{\gamma}\right]^{\gamma}$ and $\left[1 - \frac{1-u_t}{\gamma}\right]^{\gamma}$ are adaptive weights, which will further increase the penalty if an extreme token is detected with low confidence or a non-extreme token is detected with high u_t.

To implement this loss function, extreme events need to be defined first. The Transformer operates in the latent space of VQGAN and the extreme events are defined by the area-averaged precipitation accumulation. We can classify the tokens into extreme or non-extreme, since each token has its corresponding area with different rainfall intensity levels on the radar precipitation field. The extreme tokens can be decoded to high-intensity rainfall patterns and have low occurence in the training dataset. The output of the autoregressive Transformer is a probability distribution, so the extreme indicator u_t can be computed by summing up the corresponding probabilities over the extreme tokens. By substituting the probability u_t into equation (7), the EVL can be computed and added to the loss. In this way, instead of purely relying on the data, which has a limited number of extreme events, the probability of extremes can be approximated based on extreme value theory by assuming that the area-averaged precipitation accumulation follows a heavy-tailed distribution.

3. EXPERIMENTS AND RESULTS

Following the objectives of this study, two experiments are conducted: Pixel-level nowcasting evaluation and catchment-level extreme-event forecasting evaluation. Based on the definition of extreme precipitation events, events with catchment-averaged precipitation within the largest 1% are selected. Events from 2008 to 2014 are used for training, events from 2015 to 2018 are used for validation, and events from 2019 to 2021 are used for testing the models. In this way, we select 357 nationwide events in the Netherlands, corresponding to 3927 events in the catchments, for testing. As stated in Section 2.2, we consider three different loss functions: the cross entropy loss (CE), weighted cross entropy loss (WEC), and the extreme value loss (EVL). For EVL, γ is adjusted between 0.5 and 2.0, and is set to 1.0 for optimal performance.

The model output and the nowcasting results for the whole study area are evaluated in the first part. The evaluation is based on various common metrics for nowcasting, including mean absolute error (MAE), Pearson correlation score (PCC), critical success index (CSI), false alarm rate (FAR) and fractional skill score (FSS). Two

Metrics/Models	CE	PySTEPs	WCE	EVL
PCC (↑)	0.205	0.219	0.216	0.210
MAE (↑)	0.802	0.798	0.922	0.926
CSI (1mm) (↑)	0.214	0.250	0.276	0.283
CSI (8mm) (↑)	0.004	0.008	0.004	0.006
FAR (1mm) (↓)	0.533	0.617	0.605	0.618
FAR (8mm) (↓)	0.318	0.592	0.423	0.386
FSS (1km) (↑)	0.330	0.375	0.417	0.419
FSS (10km) (↑)	0.404	0.467	0.500	0.478
FSS (20km) (↑)	0.458	0.522	0.558	0.516

Table 2. Pixel-level evaluation results for nowcasting of 3-hour averages, averaged over 357 nationwide events.

thresholds (1 and 8 mm) are used for CSI and FAR and three spatial scales (1, 10 and 20 km) are used for FSS.

In the second part, to forecast catchment-level extreme events, the 3-hour catchment average precipitation accumulation is estimated from the nowcasting result, which is then compared with the corresponding extreme threshold. Each catchment-level event is classified into one of the four cases (true/false positive/negative). Finally, we evaluate the extreme event detection ability using various common metrics, including hit rate (HR), false alarm rate (FA), critical success index (CSI) and false alarm ratio (FAR).

For both evaluations, PySTEPS is used as a benchmark. PySTEPS is an open-source Python framework for ensemble precipitation nowcasting [18] and is considered as the state-of-the-art conventional (optical flow) nowcasting method [6]. In the experiments, PySTEPS is configured in ensemble mode. For both PySTEPS and the proposed deep learning model, the nowcasting output is the average of 5 ensemble members. PySTEPS also has the same input as the deep learning model, i.e., radar maps with shape of 256×256 and time intervals of 30 minutes.

3.1. Evaluation of Nowcasting Skill

We compare the output of the models with the ground truth precipitation fields. Table 2 summarizes the scores for the prediction of 3-hour averages. Two conclusions can be drawn based on the table: First, in general, our proposed models exhibit comparable nowcasting performance to PySTEPS. This proves that our proposed model architecture is suitable for the nowcasting task. Second, by using class weights and extreme value loss, the nowcasting performance shows clear improvement over the baseline cross entropy loss, which validates the usefulness of these two approaches.

3.2. Forecasting of Extreme Events

To forecast catchment-level extreme events, the catchment areas are cropped from the precipitation fields produced by different models and compared with extreme thresholds. The extreme threshold for the ground truth data is based on our definition of extreme events, while the threshold for the detection of extremes events are adjusted so that the models have the same HR of 0.8. The results are shown in Table 3. Moreover, we have swept the threshold on the output of the detectors, leading to the ROC curves as shown in Figure 3.

Based on the results, several conclusions can be drawn: First, the proposed models outperform PySTEPS for HR $\geq 60\%$, since for the same hit rate, PySTEPS has higher FA/FAR and lower CSI. This proves the effectiveness of the proposed model architecture.

Models/Metrics	HR (↑)	FA (↓)	FAR (↓)	CSI (↑)
PySTEPS	0.8	0.3627	0.6089	0.3532
Cross Entropy (CE)	0.8	0.3391	0.5970	0.3673
Weighted CE (WCE)	0.8	0.3521	0.5848	0.3667
CE + EVL	0.8	0.3159	0.5819	0.3782

Table 3. Catchment-level evaluation results for extreme-event forecasting, averaged over 3927 catchment-level events.

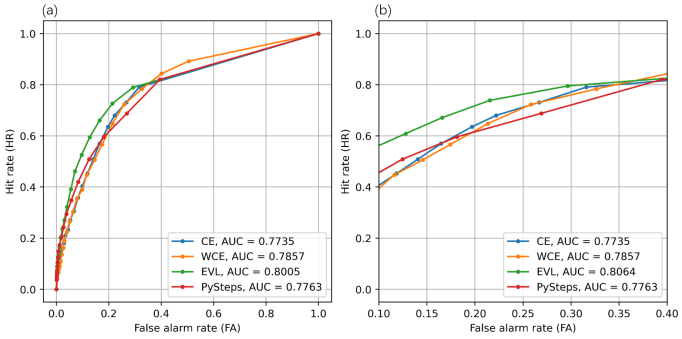

Fig. 3. (a) The ROC curves for 3-hour extreme precipitation event detection. (b) Close-up of the ROC curves.

Second, CE and WCE are both outperformed by EVL. For CE, the imbalanced occurrence of the tokens in the training set may lead to an underfitting problem and may bias the probability distribution toward the majority class. For WCE, the weights are estimated from the occurence in the training set. This choice assumes that the distribution in the dataset reflects the actual distribution. However, the scarcity of extreme samples makes it hard to represent the extreme (right tail) part of the distribution. The EVL essentially also adjusts the weights of the tokens. The weights are not purely based on the data, but also based on the approximation (6) of the tail distribution of precipitation data. The result proves that the proposed EVL is promising for modelling extreme events.

Third, the ROC curve in Figure 3 further supports our conclusions. In terms of the area under curve (AUC), the WCE and EVL both outperform CE. In terms of the complete ROC curve, the difference between the models is less obvious. However, if we limit the FA and HR within reasonable ranges (as shown in Figure 3(b)), the EVL clearly shows a better performance.

4. CONCLUSION

In this paper, we proposed "VQGAN + Transformer" for precipitation nowcasting and extreme precipitation event forecasting. Compared with typical deep learning tasks, one difficulty of the nowcasting task is the highly imbalanced distribution of the precipitation data. To address this challenge, we explored applying different loss functions, including WCE and EVL, for better modeling of extreme events. Based on the numerical results, we can conclude that the proposed model is suitable for nowcasting and can show comparable overall nowcasting performance to PySTEPS. Second, in terms of extreme event forecasting, WCE shows similar performance as PySTEPS, while EVL achieves clear improvement over other models, indicating its promising potential in modeling and predicting extreme events.

5. REFERENCES

[1] Rachel Prudden, Samantha Adams, Dmitry Kangin, Niall Robinson, Suman Ravuri, Shakir Mohamed, and Alberto Arribas, "A review of radar-based nowcasting of precipitation and applicable machine learning techniques," *arXiv preprint arXiv:2005.04988*, 2020.

[2] Ruben Imhoff, C. Brauer, Klaas-Jan van Heeringen, Remko Uijlenhoet, and Albrecht Weerts, "Large-sample evaluation of radar rainfall nowcasting for flood early warning," *Water Resources Research*, vol. 58, 03 2022.

[3] Xingjian Shi, Zhourong 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*, vol. 28, 2015.

[4] Vadim Lebedev, Vladimir Ivashkin, Irina Rudenko, Alexander Ganshin, Alexander Molchanov, Sergey Ovcharenko, Ruslan Grokhovetskiy, Ivan Bushmarinov, and Dmitry Solomentsev, "Precipitation nowcasting with satellite imagery," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2680–2688.

[5] Kevin Trebing, Tomasz Staczyk, and Siamak Mehrkanoon, "Smaat-unet: Precipitation nowcasting using a small attention-unet architecture," *Pattern Recognition Letters*, vol. 145, pp. 178–186, 2021.

[6] Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al., "Skilful precipitation nowcasting using deep generative models of radar," *Nature*, vol. 597, no. 7878, pp. 672–677, 2021.

[7] JR Jing, Qian Li, XY Ding, NL Sun, Rong Tang, and YL Cai, "Aenn: A generative adversarial neural network for weather radar echo extrapolation," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, pp. 89–94, 2019.

[8] John S Marshall and W. Mc K. Palmer, "The distribution of raindrops with size," *J. meteor.*, vol. 5, pp. 165–166, 1948.

[9] RO Imhoff, CC Brauer, A Overeem, AH Weerts, and R Uijlenhoet, "evaluation of radar rainfall nowcasting techniques on 1,533 events," *Water Resources Research*, vol. 56, no. 8, pp. e2019WR026723, 2020.

[10] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan, "Nuwa: Visual synthesis pre-training for neural visual world creation," *arXiv preprint arXiv:2111.12417*, 2021.

[11] Patrick Esser, Robin Rombach, and Bjorn Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12873–12883.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[13] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al., "Improving language understanding by generative pre-training," 2018.

[14] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas, "Videogpt: Video generation using vq-vae and transformers," *arXiv preprint arXiv:2104.10157*, 2021.

[15] Daizong Ding, Mi Zhang, Xudong Pan, Min Yang, and Xiangnan He, "Modeling extreme events in time series prediction," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2019, KDD '19, p. 1114–1122, Association for Computing Machinery.

[16] Janos Galambos, "The asymptotic theory of extreme order statistics," Tech. Rep., 1978.

[17] Rym Worms, "Propriété asymptotique des excès additifs et valeurs extrêmes: le cas de la loi de gumbel," *Comptes Rendus de l'Academie des Sciences Series I Mathematics*, vol. 5, no. 327, pp. 509–514, 1998.

[18] Seppo Pulkkinen, Daniele Nerini, Andrés A Pérez Hortal, Carlos Velasco-Forero, Alan Seed, Urs Germann, and Loris Foresti, "Pysteps: an open-source python library for probabilistic precipitation nowcasting (v1. 0)," *Geoscientific Model Development*, vol. 12, no. 10, pp. 4185–4219, 2019.