# Extreme Precipitation Nowcasting using Transformer-based Generative Models

**Cristian Meo**[1*] **Ankush Roy**[1*] **Mircea Lică**[1*] **Junzhe Yin**[1]

**Zeineb Bou Cher**[1] **Yanbo Wang**[1] **Ruben Imhoff**[2] **Remko Uijlenhoet**[1]

**Justin Dauwels**[1]

## Abstract

This paper presents an innovative approach to extreme precipitation nowcasting by employing Transformer-based generative models, namely NowcastingGPT with Extreme Value Loss (EVL) regularization. Leveraging a comprehensive dataset from the Royal Netherlands Meteorological Institute (KNMI), our study focuses on predicting short-term precipitation with high accuracy. We introduce a novel method for computing EVL without assuming fixed extreme representations, addressing the limitations of current models in capturing extreme weather events. We present both qualitative and quantitative analyses, demonstrating the superior performance of the proposed NowcastingGPT-EVL in generating accurate precipitation forecasts, especially when dealing with extreme precipitation events. The code is available at `https://github.com/Cmeo97/NowcastingGPT`.

## 1 Introduction

The advent of climate change has escalated the frequency of intense rainfall events across various regions worldwide, leading to considerable societal and infrastructural impacts (Alfieri et al., 2017; Martinkova & Kysely, 2020; Klocek et al., 2021; Czibula et al., 2021; Malkin Ondík et al., 2022). Consequently, the ability to accurately forecast short-term shifts in rainfall patterns is gaining importance, attracting a growing body of research focus (Shi et al., 2015; Trebing et al., 2021; Luo et al., 2021; Liu et al., 2022). The field of precipitation nowcasting, which involves predicting rainfall changes within a six-hour window, plays a crucial role in enabling timely responses to these rapid meteorological variations (Veillette et al., 2020a; Malkin Ondík et al., 2022; Yang & Mehrkanoon, 2022; Prudden et al., 2020). In the context of escalating climate change impacts, the field of precipitation nowcasting is increasingly vital for mitigating the adverse effects of intense rainfall events. This research area empowers the development of advanced forecasting models that can provide accurate, short-term rainfall predictions. Such capabilities are essential for proactive disaster management and climate resilience strategies, enabling communities and infrastructure planners to prepare for and respond to extreme weather events more effectively, thereby contributing to meaningful efforts in addressing the climate crisis.

## 2 Related Works

Conventional nowcasting techniques, exemplified by frameworks such as PySTEPS (Pulkkinen et al., 2019), adopt the ensemble-based methodology reminiscent of Numerical Weather Prediction (NWP) to incorporate uncertainty while modeling precipitation dynamics through the lens of the advection equation (Ravuri et al., 2021). On the other hand, Deep learning-based approaches, leveraging extensive datasets of radar observations, can

[*] Equal Contribution, [1] Delft University of Technology, Netherlands [2] Deltares, Netherlands. Corresponding authors: c.meo@tudelft.nl, A.Roy-8@student.tudelft.nl, M.T.Lica@student.tudelft.nl.

be trained without the constraints of predefined physical assumptions, significantly enhancing forecast accuracy (Ravuri et al., 2021). In the last few years, precipitation nowcasting using deep learning models has been cast as a video prediction problem (Bi et al., 2023; Bai et al., 2022; Luo et al., 2021; Liu et al., 2022), where given an input spatio-temporal sequence of $N$ frames $\boldsymbol{x}_{\text{in}} \in \mathbb{R}^{N \times H \times W \times C}$, $H, W$ denote the spatial resolution and $C$ represents the image channels or the different type of measurements (e.g., radar, heat maps, etc), the goal is to predict the next $M$ frames $\boldsymbol{x}_{\text{out}} \in \mathbb{R}^{M \times H \times W \times C}$. Among the most notable advancements in the field, Generative Adversarial Networks (GAN; Goodfellow et al., 2014) have emerged as a powerful approach, exemplified by methods such as DGMR Ravuri et al. (2021), which employs both spatial and temporal discriminators to ensure the fidelity of generated sequences to the ground truth. Moreover, Transformer-based strategies Vaswani et al. (2017) leverage an Autoregressive Transformer (AT) to model the hidden dynamics of precipitation maps (Jin et al., 2024; Bi et al., 2023). For instance, Bi et al. (2023) employs Nuwä (Wu et al., 2022), an AT that uses a sparse attention mechanism, namely 3DNA (Wu et al., 2022), to adeptly capture the complexities of precipitation dynamics. Moreover, Bi et al. (2023) regularizes the hidden dynamics incorporating an Extreme Values Loss (EVL) to effectively model and predict extreme precipitation events, which are notoriously difficult to represent and predict. Although these models have improved in terms of prediction capabilities, they present critical drawbacks. Firstly, the prediction quality degrades very quickly, resulting in predicted sequences that are inconsistent over time. Secondly, the time required to generate the predicted sequences is extremely high, which is a critical problem considering that nowcasting predictions are supposed to predict the very next future. For instance, Nuwä-EVL (Bi et al., 2023) takes over 5 minutes to predict the next precipitation maps on a Nvidia RTX A6000. Furthermore, predicting and representing extreme precipitation events is still very challenging for all the proposed models. Although Bi et al. (2023) uses an EVL as a regularizer, it assumes a predefined set of representations that should embed the extreme events features, assuming that the extreme features never change during training. Since the topology of the hidden space changes during training, we believe that using fixed representations is a wrong inductive bias. In this work, we propose NowcastingGPT, which follows VideoGPT framework (Yan et al., 2021), employing a Vector Quantized-Variational AutoEncoder (VQ-VAE) (Van Den Oord et al., 2017) to extract discrete tokens and an Autoregressive Transformer (Esser et al., 2021) to model the hidden dynamics. Moreover, we propose a novel approach to correctly compute the EVL regularization without assuming any fixed extreme representation. Moreover, we benchmark TECO (Yan et al., 2023), an efficient transformer-based video prediction model that generates temporally consistent frames, on the precipitation nowcasting task. Finally, we present both qualitative and quantitative comparisons of the considered models.

## 3 Methodology

Video prediction tasks, at their core, involve forecasting the future frames of a video sequence based on past observations, akin to predicting the next scenes in a dynamic storyline. This challenge extends naturally to nowcasting, where the goal is predicting satellite imagery or radar maps, capturing the evolution of environmental and weather conditions over time. Both domains share the fundamental task of modeling and anticipating the progression of complex, time-varying patterns, making techniques developed for video prediction highly relevant and applicable to the realm of nowcasting.

### 3.1 Nowcasting as Video Prediction

Video prediction tasks are known for their sample inefficiency, which poses significant challenges in learning accurate and reliable models. To address this, recent advancements have introduced spatio-temporal state space models, which typically consist of a feature extraction component coupled with a dynamics prediction module. These models aim to understand and predict the evolution of video frames by capturing both spatial and temporal relationships. Notable examples include Nuwä (Wu et al., 2022) and VideoGPT (Yan et al., 2021) which, leveraging the space-efficient VQ-VAE feature extraction, and the powerful sequence modeling capabilities of Autoregressive Transformers, can achieve a deeper understanding of the underlying video dynamics, leading to more accurate predictions of future frames. We define the video prediction backbone of the proposed nowcasting model following the VideoGPT framework, using a VQ-VAE as a feature extractor and an Autoregressive Transformer (Esser et al., 2021) to learn the latent space dynamics and predict the future precipitation maps. A detailed description of the NowcastingGPT model, depicted in Fig. 1, can be found in appendix 6.

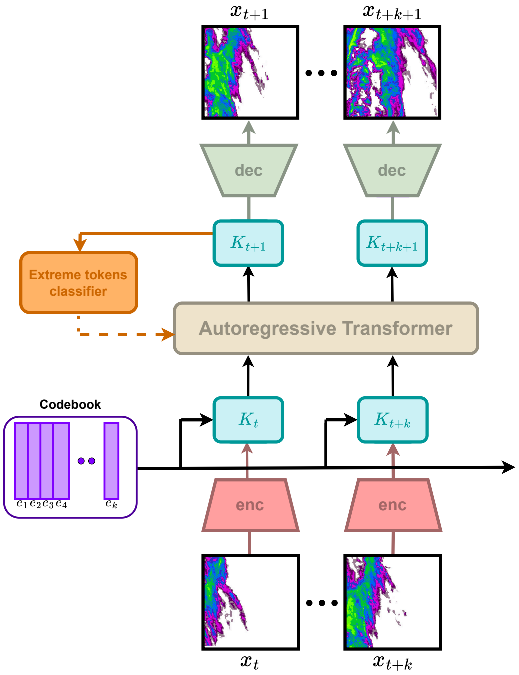

Figure 1: The image shows the NowcastingGPT-EVL model architecture. The VQ-VAE Encoder and Decoder are depicted in red and green respectively. The Extreme tokens classifier is depicted in orange, it takes the predicted tokens as input from the transformer and outputs the probabilities $u_t$ used in the EVL loss. The dashed line indicates that the output of the Classifier is only used to optimize the transformer and not as input.

### 3.2 Extreme Value Loss Regularization

When dealing with imbalanced data, the standard cross-entropy loss often falls short, particularly when classifying extreme events. To address this, the Extreme Value Loss (EVL) has been introduced as a more effective alternative, designed to balance the disparities between extreme and non-extreme cases in time series data Ding et al. (2019):

$$\mathrm{EVL}(u_t, v_t) = -\beta_1 \left[1 - \frac{u_t}{\gamma}\right]^{\gamma} v_t \log(u_t) - \beta_0 \left[1 - \frac{1-u_t}{\gamma}\right]^{\gamma} (1 - v_t)\log(1 - u_t), \quad (1)$$

where $v_t$ represents the ground truth labels (extreme/not extreme), $u_t$ the predicted probabilities, and $\gamma$, a hyperparameter of the Generalized Extreme Value (GEV) distribution. By incorporating $\beta_0$ and $\beta_1$, which reflect the proportions of non-extreme and extreme events, EVL effectively balances the learning process. When regularizing an Autoregressive Transformer the EVL enhances the model's ability to predict and represent extreme events. To this end, we define a classifier that dynamically predicts extreme labels. As a result, we can use the EVL to regularize the Autoregressive Transformer learning behavior and improve its ability to capture extreme phenomena in data sequences. A detailed description of the classifier can be found in appendix 6.4.3, while the full derivation of the EVL loss can be found in appendix 6.5.

# 4 Experiments

In this section, we design empirical experiments to understand the performance of NowcastingGPT-EVL and its potential limitations by exploring the following questions: (1) Does EVL regularization improve the nowcasting performances of the proposed model? (2) How does time consistency affect downstream results? (3) Does learning extreme representations provide a more effective inductive bias compared to relying on predefined ones?

## 4.1 Dataset and Experimental setup

Our nowcasting study aims to predict precipitation patterns up to three hours into the future. This approach generates a series of six future precipitation maps, each separated by 30 minutes, conditioned on three previous precipitation maps used as input. Specifically, we use radar maps defined 256 × 256 images, which follow the approach used in (Bi et al., 2023). More details about the dataset are described in appendix 6.1. We compare the proposed model to a classic benchmark, namely Pysteps (Pulkkinen et al., 2019), a temporally consistent video prediction benchmark, TECO (Yan et al., 2023), the NowcastingGPT model which is described in the appendix 6.4.2 and Nuwä-EVL proposed by Bi et al. (2023), which uses fixed latents to represents extreme features. An in-depth description of the considered baselines can be found in appendix 6.3. To quantitatively assess the experiments we use visual fidelity metrics, such as Mean Squared Error (MSE), Mean Absolute Error (MAE) and Pearson Correlation Score (PCC), and nowcasting metrics, such as Critical Success Index (CSI), False Alarm Rate (FAR) and Fractional Skill Score (FSS). Since fidelity metrics cannot capture extreme event classification, we plot an ROC curve of the extremes to assess the considered baselines in terms of extreme classification capabilities. A detailed description of these metrics can be found in appendix 6.2. Table 1 presents a quantitative comparison between all proposed methods in terms of number of parameters, training and generation efficiency. Interestingly, TECO, showcases orders of magnitude more efficient generation time and cuts the training time by approximately 100 hours compared to its closest counterpart. Furthermore, with a generation time of 322.86 seconds, Nuwä-EVL constitutes a good indicator for the sampling efficiency of autoregressive models.

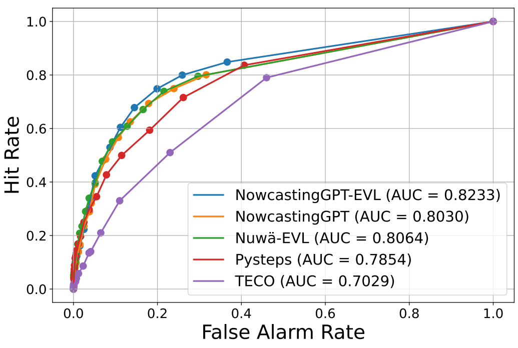

Figure 2: ROC Curve for extreme event detection. Thresholds between 0.5 and 10 for precipitation values are used to define an extreme event. NowcastingGPT-EVL had the highest AUC, outperforming all other baselines.

| | Nuwä-EVL | NowcastingGPT | PySTEPS | TECO | NowcastingGPT-EVL |
|---|---|---|---|---|---|
| Number of parameters | $772,832$ M | $402,735$ M | - | $165,960$ M | $520,374$ M |
| Training time | 672h | 240h | - | 155h | 264h |
| Generation time | 322.86s | 38.90s | 9.34s | 0.51s | 43.10s |

Table 1: Comparison of the proposed methods in terms of number of parameters, training time and generation time. Generation time refers to the time required on average to sample a sequence from the dataset defined in Section 6.1. Training time is computed in terms of GPU hours.

Table 2: Quantitative results of the proposed methods. Each value represents the average and standard deviation over the means and standard deviations of each of the 6 lead times. The description for each metric can be found in appendix 6.2. For statistically meaningful results, we consider 3 different seeds for each entry.

| | Nuwä-EVL | NowcastingGPT | PySTEPS | TECO | NowcastingGPT-EVL |
|---|---|---|---|---|---|
| PCC (↑) | 0.15 | 0.20 ± 0.002 | 0.14 | 0.10 ± 0.002 | **0.22** ± 0.002 |
| MSE (↓) | 4.85 | 3.60 ± 0.02 | 6.22 | 3.65 ± 0.008 | **3.45** ± 0.02 |
| MAE (↓) | 1.00 | 0.72 ± 0.005 | 0.93 | **0.68** ± 0.001 | 0.69 ± 0.005 |
| CSI(1mm) (↑) | **0.23** | 0.21 ± 0.002 | 0.21 | 0.07 ± 0.001 | 0.22 ± 0.002 |
| CSI(2mm) (↑) | **0.13** | 0.11 ± 0.001 | 0.12 | 0.03 ± 0.001 | 0.12 ± 0.001 |
| CSI(8mm) (↑) | 0.008 | 0.005 ± 0.0005 | **0.01** | 0.001 ± 0.0009 | 0.009 ± 0.0005 |
| FAR(1mm) (↓) | 0.61 | 0.59 ± 0.002 | **0.55** | 0.69 ± 0.002 | 0.59 ± 0.002 |
| FAR(2mm) (↓) | 0.76 | 0.71 ± 0.0007 | **0.70** | 0.78 ± 0.004 | 0.71 ± 0.0007 |
| FAR(8mm) (↓) | 0.85 | 0.59 ± 0.003 | 0.89 | **0.49** ± 0.006 | 0.52 ± 0.003 |
| FSS(1km) (↑) | 0.35 | 0.49 ± 0.003 | 0.32 | 0.49 ± 0.003 | **0.52** ± 0.003 |
| FSS(10km) (↑) | 0.42 | 0.55 ± 0.004 | 0.41 | 0.46 ± 0.003 | **0.58** ± 0.004 |
| FSS(20km) (↑) | 0.48 | 0.59 ± 0.004 | 0.47 | 0.42 ± 0.003 | **0.62** ± 0.004 |
| FSS(30km) (↑) | 0.52 | 0.62 ± 0.004 | 0.51 | 0.37 ± 0.002 | **0.65** ± 0.004 |

### 4.2 Experimental Results

We test the performance of the proposed models by using the extreme precipitation test set described in appendix 6.1. Table 2 showcases the effectiveness of these methods against a series of metrics that asses the quality and validity of the predictions. The proposed NowcastingGPT-EVL outperforms the other models on the majority of metrics and close second on the rest. The ROC curve in Figure 2 demonstrates how NowcastingGPT-EVL outperforms all other methods on extreme event detection at different thresholds. Figure 4 illustrates the predicted maps of all considered baselines. While NowcastingGPT presents meaningful predictions over all time steps, Nuwä-EVL deteriorates substantially. Indeed, we believe that when Nuwä-EVL extreme representations get updated by the AT, the VQ-VAE is not able to recognize the extreme latents anymore, which by design are supposed to be fixed, predicting images that do not resemble the ground truth maps semantics. Remarkably, the graphs presented in Appendix 6.6 demonstrate that TECO achieves results on par with other methods, despite having fewer parameters and a more efficient sampling time, and exhibits superior temporal consistency compared to alternative approaches.

## 5 Conclusion & Discussion

This work proposes NowcastingGPT-EVL, a video prediction model regularized using an EVL regularizer, validating the efficacy of using EVL for nowcasting extreme precipitation events. Our findings reveal that the proposed model outperforms existing methods in various downstream metrics, providing more accurate predictions. The study highlights the importance of addressing data imbalances and the dynamic nature of extreme events in model training. As future work, we aim to assess the prediction capabilities of the different models on an existing and widely used benchmark dataset (e.g., SEVIR (Veillette et al., 2020b)). The successful application of NowcastingGPT-EVL underscores the potential of Transformer-based models in enhancing predictive capabilities for critical meteorological forecasting tasks, paving the way for future advancements in the field.

## References

Lorenzo Alfieri, B Bisselink, Francesco Dottori, Gustavo Naumann, A. P. J. De Roo, Péter Salamon, Klaus Wyser, and Luc Feyen. Global projections of river flood risk in a warmer world. *Earth's Future*, 5, 2017. URL https://api.semanticscholar.org/CorpusID:42772267.

Cong Bai, Feng Sun, Jinglin Zhang, Yi Song, and Shengyong Chen. Rainformer: Features extraction balanced network for radar-based precipitation nowcasting. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022. URL https://api.semanticscholar.org/CorpusID:248132089.

Haoran Bi, Maksym Kyryliuk, Zhiyi Wang, Cristian Meo, Yanbo Wang, Ruben Imhoff, Remko Uijlenhoet, and Justin Dauwels. Nowcasting of extreme precipitation using deep generative models. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10094988.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

Shengyu Chen, Nasrin Kalanat, Simon Topp, Jeffrey Sadler, Yiqun Xie, Zhe Jiang, and Xiaowei Jia. Meta-transfer-learning for time series data with extreme events: An application to water temperature prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, New York, NY, USA, October 2023. ACM.

Stuart Coles. *An introduction to statistical modeling of extreme values*. Springer Series in Statistics. Springer-Verlag, 2001. ISBN 1-85233-459-2.

Gabriela Serban Czibula, Andrei Mihai, Alexandra-Ioana Albu, István Gergely Czibula, Sorin Burcea, and Abdelkader Mezghani. Autonowp: An approach using deep autoencoders for precipitation nowcasting based on weather radar reflectivity prediction. *Mathematics*, 2021. URL https://api.semanticscholar.org/CorpusID:238018677.

Laurens De Haan and Ana Ferreira. *Extreme Value Theory*. Springer Series in Operations Research and Financial Engineering. Springer Science+Business Media, January 2006.

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*, KDD '19, pp. 1114–1122, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330896. URL https://doi.org/10.1145/3292500.3330896.

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*, pp. 12873–12883, 2021.

Ramazan Gençay and Faruk Selçuk. Extreme value theory and Value-at-Risk: Relative performance in emerging markets. *Int. J. Forecast.*, 20(2):287–303, April 2004.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

RO Imhoff, CC Brauer, A Overeem, AH Weerts, and R Uijlenhoet. Spatial and temporal evaluation of radar rainfall nowcasting techniques on 1,533 events. *Water Resources Research*, 56(8):e2019WR026723, 2020.

Qizhao Jin, Xinbang Zhang, Xinyu Xiao, Ying Wang, Shiming Xiang, and Chunhong Pan. Preformer: Simple and efficient design for precipitation nowcasting with transformers. *IEEE Geoscience and Remote Sensing Letters*, 21:1–5, 2024. doi: 10.1109/LGRS.2023.3325628.

Sylwester Klocek, Haiyu Dong, Matthew Dixon, Panashe Kanengoni, Najeeb Kazmi, Pete Luferenko, Zhongjian Lv, Shikhar Sharma, Jonathan A. Weyn, and Siqi Xiang. Ms-nowcasting: Operational precipitation nowcasting with convolutional lstms at microsoft weather. *ArXiv*, abs/2111.09954, 2021. URL `https://api.semanticscholar.org/CorpusID:244463010`.

Jie Liu, Lei Xu, and Nengcheng Chen. A spatiotemporal deep learning model st-lstm-sa for hourly rainfall forecasting using radar echo images. *Journal of Hydrology*, 2022. URL `https://api.semanticscholar.org/CorpusID:247602986`.

Chuyao Luo, Xinyue Zhao, Yuxi Sun, Xutao Li, and Yunming Ye. Predrann: The spatiotemporal attention convolution recurrent neural network for precipitation nowcasting. *Knowl. Based Syst.*, 239:107900, 2021. URL `https://api.semanticscholar.org/CorpusID:245591327`.

Irina Malkin Ondík, Lukáš Ivica, Peter Šišan, Ivan Martynovskyi, David Šaur, and Ladislav Gaál. A concept of nowcasting of convective precipitation using an x-band radar for the territory of the zlín region (czech republic). In *Computer Science On-line Conference*, pp. 499–514. Springer, 2022.

Marta Martinkova and Jan Kysely. Overview of observed clausius-clapeyron scaling of extreme precipitation in midlatitudes. *Atmosphere*, 11(8):786, 2020.

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.

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*, 12(10):4185–4219, 2019.

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*, 597(7878):672–677, 2021.

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*, 28, 2015.

R Sluiter. *Interpolation methods for the climate atlas.* KNMI De Bilt, The Netherlands, 2012. URL `https://cdn.knmi.nl/knmi/pdf/bibliotheek/knmipubTR/TR335.pdf`.

Kevin Trebing, Tomasz Stanczyk, and Siamak Mehrkanoon. Smaat-unet: Precipitation nowcasting using a small attention-unet architecture. *Pattern Recognition Letters*, 145: 178–186, 2021.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Marc Veillette, Siddharth Samsi, and Christopher J. Mattioli. Sevir : A storm event imagery dataset for deep learning applications in radar and satellite meteorology. In *Neural Information Processing Systems*, 2020a. URL `https://api.semanticscholar.org/CorpusID:227222587`.

Mark Veillette, Siddharth Samsi, and Chris Mattioli. Sevir : A storm event imagery dataset for deep learning applications in radar and satellite meteorology. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 22009–22019. Curran Associates, Inc., 2020b. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/fa78a16157fed00d7a80515818432169-Paper.pdf`.

Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pre-training for neural visual world creation. In *European conference on computer vision*, pp. 720–736. Springer, 2022.

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

Wilson Yan, Danijar Hafner, Stephen James, and Pieter Abbeel. Temporally consistent transformers for video generation. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 39062–39098, 2023.

Yimin Yang and Siamak Mehrkanoon. Aa-transunet: Attention augmented transunet for nowcasting tasks. *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 01–08, 2022. URL `https://api.semanticscholar.org/CorpusID:246705912`.

# 6 Appendix

## 6.1 Dataset

The reflectivity measurements in the KNMI (Sluiter, 2012) dataset allows for the estimation of rainfall rates through the application of a Z-R transformation, enabling a nuanced river catchment-level analysis to evaluate the model's effectiveness in real-world scenarios. Ideally, extreme rainfall events are identified based on the distribution of the highest annual rainfall amounts. However, given the limited span of our dataset, which encompasses only 14 years, the dataset provides an insufficient quantity of annual maximums for effective model training and evaluation. Consequently, we have broadened the criteria for what constitutes extreme rainfall. Within this study, an event is classified as extreme if the average precipitation over a three-hour period within a catchment area ranks within the top 1% of all observations recorded from 2008 to 2021. This adjustment allows for a more feasible and statistically sound basis for distinguishing significant precipitation events during the study period. The training dataset consists of 30632 sequences of images with each sequence consisting of 9 images (T-60, T-30, T, T+30, T+60, T+90, T+120, T+150, T+180 minutes) spanning from 2008-2014. The validation dataset consists of 3560 sequences of images with the same sequence length from year 2015-2018. The testing dataset utilised to evaluate the performance of the different models in this study, consists of 357 nationwide extreme events from 2019-2021, corresponding to 3927 events in the catchment regions.

## 6.2 Metrics

The effectiveness of any predictive model is critically assessed through objective metrics that encapsulate its performance capabilities. In our endeavor to evaluate the impact of integrating the EVL regularization, we utilize a comprehensive set of performance metrics:

- *Mean Absolute Error (MAE)*: MAE quantifies the average magnitude of errors in the predictions. It's computed as the mean of the absolute differences between the predicted values and the actual observations, offering a clear and intuitive metric for prediction accuracy.

- *Mean Squared Error (MSE)*: MSE measures the average of the squares of the errors between the predicted and actual values, providing a more sensitive metric that penalizes larger errors more severely than MAE.

- *Pearson Correlation Coefficient (PCC)*: PCC assesses the linear correlation between the predicted and observed datasets, yielding a value between -1 and 1, where 1 indicates perfect positive correlation, -1 indicates perfect negative correlation, and 0 signifies no linear correlation.

- *Critical Success Index (CSI)*: CSI is utilized to evaluate the precision of forecasted events, particularly the successful prediction of specific events. This study examines CSI at two distinct precipitation thresholds: 1mm for light precipitation and 8mm for heavy precipitation, thus catering to varying intensities of rainfall.

- *False Alarm Rate (FAR)*: FAR is calculated as the proportion of false positive predictions relative to the total number of positive forecasts (false positives plus true positives), offering insight into the model's tendency to incorrectly predict events that do not occur.

- *Fractional Skill Score (FSS)*: FSS measures the model's forecast accuracy at specific spatial scales, facilitating an understanding of how well the model performs both locally and over broader areas. In this study, FSS is evaluated at 1km, 10km, 20km and 30km scales to discern the model's effectiveness at varying geographical extents.

While both MAE and MSE loss quantify the quality of the predictions, they are not able to capture the model's capability to detect extreme events. Thus, we make use of a Receiver Operating Characteristic (ROC) curve to asses hit rate detection of extreme precipitation events. The curve is constructed using a set of thresholds that are used to define an event.

### 6.3 Baselines Comparison

Motivated by the overall inefficiency in nowcasting methods, we consider TECO (Yan et al., 2023) as a point of reference in benchmarking both training and sampling time of Transformer-based nowcasting models. TECO aims to increase sampling efficiency by replacing the common autoregressive prior with a masked token prediction objective, introduced by Chang et al. (2022). Using the discrete tokens from a VQ-VAE, the model learns to predict a randomly generated mask sampled at each timestamp, allowing for orders of magnitude improvement in sampling speed. Moreover, TECO manages to drastically decrease training time by using DropLoss, a trick that allows the model to consider only a subset of the frames that compose the video. Moreover, to be consistent with the literature, we consider PySTEPS (Pulkkinen et al., 2019), a widely used numerical model for short-term precipitation predictions that achieves remarkable results in nowcasting (Imhoff et al., 2020).

### 6.4 NowcastingGPT-EVL Description

In this section, we describe the used model. Figure 1 illustrates the model architecture. The following subsections describe the three main components: VQ-VAE, Autoregressive Transformer, and Extreme tokens classifier.

#### 6.4.1 Vector Quantized Variational Autoencoder

The Vector Quantized Variational AutoEncoder (VQVAE) (Van Den Oord et al., 2017) introduces a novel approach by utilizing Vector Quantization to encode inputs into discrete latent representations, moving away from continuous feature representations. This method is effective in capturing the complex, multi-dimensional features of data. VQVAE operates on an encoder-decoder framework with a discrete codebook, where the encoder compresses input data into a discrete set of codes, preserving essential features through a reduction in spatial dimensions and an increase in feature channels. The decoder then reconstructs the input from these codes, aiming for a close approximation to the original, thereby enabling efficient and structured data representation suitable for tasks like image reconstruction. The encoder consists of 5 downsampling layers each containing 2 ResNet blocks, thus reducing the spatial dimension of the input to the following resolutions: $128 \rightarrow 64 \rightarrow 32 \rightarrow 16 \rightarrow 8$. Furthermore, the last stage of the encoder includes an attention block used to capture the relationships between features before the quantization step. In order to obtain the reconstructed image from the discrete codes, we use a decoder that mirrors the structure of the encoder.

To facilitate the training of the VQVAE model, a set of distinct loss functions are harnessed and subjected to optimization. These loss functions encompass the reconstruction loss, the commitment loss, and the perceptual loss.

$$\mathcal{L}(E, D, \mathcal{Z}) = \|x - \hat{x}\|_2^2 + \|\text{sg}[E(x)] - z_{\mathbf{q}}\|_2^2 \quad + \|\text{sg}\left[z_{\mathbf{q}}\right] - E(x)\|_2^2 + \mathcal{L}_{\text{perceptual}}(x, \hat{x}), \tag{2}$$

where $\hat{z} = E(x) \in \mathbb{R}^{h \times w \times n_z}$ represents the encoded image while $\hat{x} = D(z_q)$ is the reconstructed image using $z_{\mathbf{q}}$. We obtain $z_{\mathbf{q}}$ using an element-wise quantization $q(.)$ of each spatial code $\hat{z}_{ij} \in \mathbb{R}^{n_z}$ given by

$$z_{\mathbf{q}} = \mathbf{q}(\hat{z}) := \left( \underset{z_k \in \mathcal{Z}}{\arg\min} \|\hat{z}_{ij} - z_k\| \right) \in \mathbb{R}^{h \times w \times n_z}.$$

#### 6.4.2 Autoregressive Transformer

In order to model the dynamics between consecutive precipitation maps, we use an Autoregressive Transformer. For training the model, we utilize the ground truth precipitation maps which are quantized into $\mathbf{z}_q = q(\mathbf{E}(x))$, generating a sequence $\mathbf{s} \in \{0, \ldots, |\mathbf{Z}| - 1\}^{h \times w}$, corresponding to the respective indices of the VQVAE codebook. Subsequently, these indices are transformed into continuous vector representations using an embedder. In order to provide sequence order information to the Transformer, the representations are augmented with positional embeddings. These are then processed by the Transformer, which outputs the logits generated by the head module. These logits represent the probability of using a specific token and are used to compute a cross-entropy loss. The loss compares the predicted probabilities given by the model with the actual token probabilities:

$$\mathcal{L}_{\text{Transformer}} = \mathbb{E}_{x\sim p(x)}[-\log \prod_{i=1}^{N} p(\mathbf{s}_i|\mathbf{s}_{<i})] \tag{3}$$

which, given a sequence of indices $\mathbf{s}_{<i}$, the Transformer is trained to predict the distribution of the consecutive indices $\mathbf{s}_i$. The AT employs a causal attention mechanism, where the non-causal entries of $QK^T$, those below the diagonal of the attention matrix, are set to $-\infty$. As a result, the attention mechanism accesses only previously seen or current tokens when predicting the next one in a sequence, enabling efficient and context-sensitive output production. We use the architecture described above to define our ablation model NowcastingGPT.

The Autoregressive Transformer for NowcastingGPT-EVL has the EVL loss function incorporated in it so the overall loss function for the AR transformer is given as:

$$\mathcal{L}_{\text{Transformer(NowcastingGPT-EVL)}} = \mathcal{L}_{\text{Transformer}} + \lambda[\text{EVL}(u_t, v_t)]. \tag{4}$$

The value of $\lambda$ in the equation above is chosen as 0.5.

### 6.4.3 Binary Classifier

For the classification of the tokens into extreme or non-extreme, a transformer is incorporated along with the auto-regressive transformer. The input to this transformer are the sequence of tokens that are generated from the auto-regressive transformer during its training phase. The model has 6 layers, 1024 embedding dimension and, a total number of 8 heads. The transformer is trained using a standard binary cross entropy loss function where, the ground truth labels $v_t$ are calculated on the basis of averaged precipitation over a threshold of 5mm. In this way, all the tokens corresponding to an extreme/non-extreme event get classified along with the training of the auto-regressive transformer. The classifier generates logits for the two classes (extreme and, non-extreme) which are then passed through a softmax layer to generate probabilities. These probabilities act as the input to the EVL loss function mentioned in equation (1) for the term $u_t$. The values for $\beta_0$ and $\beta_1$ are taken as 0.95 and 0.05 respectively since, top 5% of the events are considered as extreme events. The value of $\gamma$ for EVL was set to 1, as this setting demonstrated optimal performance.

## 6.5 Mathematical Proof of the EVL loss function

As mentioned in Coles (2001), if there is a sequence of independent and identically distributed (i.i.d) random variables as $X_1, X_2, \ldots, X_n$, having marginal distribution function $F$. It is natural to regard as extreme events those of the $X_i$ that exceed some high threshold $u$. Denoting an arbitrary term in the $X_i$ sequence by $X$, it follows that a description of the stochastic behavior of extreme events is given by the conditional probability:

$$\Pr\{X > u + y \mid X > u\} = \frac{1 - F(u+y)}{1 - F(u)}, \quad y > 0. \tag{5}$$

Starting from the L.H.S we have:

$$\Pr\{X > u + y \mid X > u\},$$

and using the formula $P(x \mid y) = \frac{P(x,y)}{P(y)}$:

$$\begin{aligned}\Pr\{X > u + y \mid X > u\} &= \frac{P(X > u + y, X > u)}{P(X > u)} \\ &= \frac{P(X > u + y)}{P(X > u)}.\end{aligned}$$

Applying the formula, $P(X > x) = 1 - F(x)$ we get,

$$= \frac{1 - F(u+y)}{1 - F(u)}.$$

If the parent distribution $F$ was known to us then the distribution of threshold exceedances in equation 5 would also be known, however, that is not the case. Coles (2001) suggests the application of Extreme Value Theory (EVT) for the approximation of the distribution of maxima of long sequences when the parent population function (distribution) $F$ is unknown. For the sequence of R.Vs mentioned above (with common distribution function $F$), we use maximum order statistics to characterize extremes :

$$M_n = \max\{X_1, X_2, X_3, \ldots X_n\}, \overset{P}{\to} x^*, n \to \infty. \tag{6}$$

where $\overset{P}{\to}$ denotes convergence in probability and, $x^*$ denotes the right end point which is $x^* = \sup\{x : F(x) < 1\}$ Therefore, for a large $n$ we have :

$$P(\max(X_1, X_2, \ldots, X_n) \leqslant x) = Pr(X_1 \leqslant x, X_2 \leqslant x, X_3 \leqslant x, \ldots X_n \leqslant x), \tag{7}$$

since, they are i.i.d we can also write equation 7 as,

$$P(\max(X_1, X_2, \ldots, X_n) \leqslant x) = [\Pr(X \leqslant x)]^n = [\mathrm{F}(x)]^n.$$

Hence, from

$$[F(x)]^n \to 0 \text{ for } x < x^*$$
$$[F(x)]^n \to 1 \text{ for } x \geqslant x^*,$$

it can be said said that $[F(x)]^n$ is a degenerate function as it converges to a single point when $n$ becomes sufficiently large. To mitigate this, EVT suggests that for a sequence of constants $a_n > 0$ and real $b_n$ there is a non-degenerate distribution function $G$ stated as:

$$\lim_{n \to \infty} [F(a_n x + b_n)]^n = G(x), \tag{8}$$

where $G(x)$ is the Generalised Extreme Value distribution function (GEV). The GEV is given by:

$$G(x) = \exp\left\{-\left[1 + \xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}, \tag{9}$$

where $\mu$ is the location parameter, $\sigma$ is the scale parameter and $\xi$ is the shape parameter. Also, equation (8) can be written as:

$$\begin{aligned} & [F(a_n x + b_n)]^n \approx G(x) \\ \Longrightarrow\ & [F(x)]^n \approx G\{(x - b_n)/a_n\} \\ \Longrightarrow\ & [F(x)]^n = G^*(x), \end{aligned}$$

where $G^*$ is another member of the GEV family. Coles (2001) mentions that if equation (8) allows the approximation of $[F(a_n x + b_n)]^n$ by a member of the GEV family for large $n$, then $[F(x)]^n$ can also be approximated using a different member of the GEV family ($G^*(x)$) which has the same definition as mentioned in 9 but with different values of $\mu$, $\sigma$ and $\xi$. Therefore, we can then write :

$$[F(x)]^n \approx \exp\left\{-\left[1 + \xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}. \tag{10}$$

Taking natural logarithm on both sides,

$$n(\ln F(x)) \approx -\left[1 + \xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}.$$

For large values of $x$, a Taylor expansion implies that,

$$\ln F(x) \approx -\{1 - F(x)\}.$$

Substituting this in the above equation we get,

$$1 - F(x) \approx \frac{1}{n}\left[1 + \xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}. \tag{11}$$

Now, we substitute the above result obtained in the R.H.S of equation (5) for a large $u$ and $y > 0$ as,

$$1 - F(u) \approx \frac{1}{n}\left[1 + \xi\left(\frac{u-\mu}{\sigma}\right)\right]^{-1/\xi}$$

and,

$$1 - F(u+y) \approx \frac{1}{n}\left[1 + \xi\left(\frac{u+y-\mu}{\sigma}\right)\right]^{-1/\xi}.$$

Therefore, we can write equation (5) as:

$$\begin{aligned}\Pr\{X > u + y \mid X > u\} &\approx \frac{n^{-1}[1+\xi(u+y-\mu)/\sigma]^{-1/\xi}}{n^{-1}[1+\xi(u-\mu)/\sigma]^{-1/\xi}} \\ &= \left[1 + \frac{\xi y/\sigma}{1+\xi(u-\mu)/\sigma}\right]^{-1/\xi} \\ &= \left[1 + \frac{\xi y}{\tilde{\sigma}}\right]^{-1/\xi},\end{aligned} \tag{12}$$

where $\tilde{\sigma} = \sigma + \xi(u - \mu)$. This distribution function is known as the Generalised Pareto Distribution (GPD) function which helps in modeling observations over a large enough threshold $u$ (Peaks Over Threshold method - POT) and is written formally as :

$$H(y) = 1 - \left(1 + \frac{\xi y}{\tilde{\sigma}}\right)^{-1/\xi}, \tag{13}$$

defined on $\{y : y > 0 \text{ and } (1 + \xi y/\tilde{\sigma}) > 0\}$, where

$$\tilde{\sigma} = \sigma + \xi(u - \mu).$$

According to Coles (2001), the above relation in equation 12 implies that, if block maxima have approximating distribution $G$, then threshold excesses have a corresponding approximate distribution within the GPD family ($H$). Also, the parameters of GPD can be uniquely determined by those of the associated GEV distribution of block maxima. Moreover, the GEV distribution function and the GPD distribution function are related to each other since they have the same shape parameter $\xi$ so we can derive a rough mathematical relation between these two distribution functions as:

$$H(y) = 1 + \ln(G(y)), \tag{14}$$

for some location ($\mu$) and shape ($\sigma, \tilde{\sigma}$) parameters. This relationship is also mentioned in the paper Gençay & Selçuk (2004) which utilises EVT and Value-at-Risk for relative performance of stock market returns in emerging markets. We can rewrite equation (5) with the help of the derived results in equations (12) and (13) as:

$$\begin{aligned}\frac{1-F(u+y)}{1-F(u)} = \left[1 + \frac{\xi y}{\tilde{\sigma}}\right]^{-1/\xi} &\implies \frac{1-F(u+y)}{1-F(u)} = 1 - H(y) \\ &\implies 1 - F(u+y) \approx (1 - F(u))(1 - H(y)).\end{aligned} \tag{15}$$

This equation is the main equation for the tail approximation of observations exceeding a threshold $u$ and matches with the tail approximation equation mentioned in De Haan & Ferreira (2006) as:

$$1 - F(x) \approx (1 - F(t))\left\{1 - H_\xi\left(\frac{x-t}{f(t)}\right)\right\}, x > t, \tag{16}$$

where $H_\xi$ is the GPD function with the shape parameter $\xi$. Therefore, we use the result derived in equation (15) to derive the weights of the EVL loss function mentioned in the paper Bi et al. (2023). However, the authors utilise the GEV distribution function to define the underlying distribution of the time series data used in the paper. The goal of the paper is to predict outputs $Y_{T:T+K}$ in the future given the observations $(X_{1:T}, Y_{1:T}$ and future inputs $X_{T:T+K}$. For the sake of convenience, the authors define $X_{1:T} = [x_1, \cdots, x_T]$ and $Y_{1:T} = [y_1, \cdots, y_T]$ to denote the general input and output sequences without referring to specific sequences. Therefore, for $T$ random variables $y_1, \cdots, y_T$ i.i.d sampled from a distribution $F_Y$, the distribution of the maximum is realised using EVT as :

$$\lim_{T\to\infty} P\left\{\max\left(y_1, \cdots, y_T\right) \leq y\right\} = \lim_{T\to\infty} F^T(y) = G(y), \tag{17}$$

for some linear transformation where $G(y)$ is GEV distribution function. We can observe that equation (8) and equation (17) have the same meaning (but with different variables in their definitions). Moreover, the authors define the GEV function as:

$$G(y) = \begin{cases} \exp\left(-\left(1-\frac{1}{\gamma}y\right)^{\gamma}\right), & \gamma \neq 0, 1-\frac{1}{\gamma}y > 0 \\ \exp\left(-e^{-y}\right), & \gamma = 0, \end{cases} \tag{18}$$

where $\gamma$ is known as the extreme value index (the shape parameter) with condition $\gamma \neq 0$. It can also be observed that the definition of GEV function in equation (18) is similar to the definition mentioned in equation (9) but with $\xi = -\frac{1}{\gamma}$, $\mu = 0$ and, $\sigma = 1$. For modeling the tail distribution of the corresponding time-series data, they use equation (16) but as mentioned before, rather than using the GPD function they use the GEV distribution function to model the tail approximation. Therefore, we substitute the relationship mentioned in equation (14) as $-ln(G(y)) = 1 - H(y)$ in equation (16) and get the following result:

$$1 - F(y) \approx (1 - F(\xi))\left[-\ln G\left(\frac{y-\xi}{f(\xi)}\right)\right], y > \xi, \tag{19}$$

where $\xi$ is the threshold and, $f(\xi)$ is a scale function as mentioned in the paper Bi et al. (2023). Also, the authors define an extreme indicator sequence $V_{1:T} = [v_1, \cdots, v_T]$ as:

$$v_t = \begin{cases} 1 & y_t > \xi \\ 0 & y_t \leqslant \xi, \end{cases} \tag{20}$$

where $\xi$ is the threshold. For time step $t$ if $v_t = 0$ then the output $y_t$ is considered as a 'normal event' and if $v_t = 1$ then $y_t$ is considered as an 'extreme event'. The authors also mention a hard approximation for the term $(\frac{y-\xi}{f(\xi)})$ in equation (19) as $u_t$ which is the predicted indicator by the neural network used by them in their experiment. This can be interpreted as a normalization which restricts the values of output $y$, above and below the threshold $\xi$ between $[-1, 1]$. Therefore, considering this to be true, we can rewrite equation (19) as:

$$1 - F(y) \approx (1 - F(\xi))\left[-\ln G(u_t)\right], \tag{21}$$

Substituting the definition of GEV in equation (18) into the above equation (21) we obtain:

$$1 - F(y) \approx (1 - F(\xi))\left[1 - \frac{u_t}{\gamma}\right]^{\gamma}. \tag{22}$$

The term $1 - F(\xi)$ can be approximated as:

$$1 - F(\xi) = \Pr(y > \xi) \implies 1 - F(\xi) = \Pr(v_t = 1), \tag{23}$$

where $\Pr(v_t = 1)$ is the proportion of extreme events in the dataset. Therefore, we can rewrite equation (22) with the above substitution as:

$$1 - F(y) \approx \Pr(v_t = 1)\left[1 - \frac{u_t}{\gamma}\right]^{\gamma}. \tag{24}$$

This tail approximation is incorporated in the terms of the standard Cross Entropy (CE) function as weights to define the main EVL loss function mentioned in paper Bi et al. (2023). However, the authors in paper Bi et al. (2023) define the weight as:

$$1 - F(y) \approx (1 - \Pr(v_t = 1))\left[1 - \frac{u_t}{\gamma}\right]^{\gamma}. \tag{25}$$

Upon simplifying the term $(1 - \Pr(v_t = 1)$ we get:

$$\begin{aligned} & 1 - \Pr\left(v_t = 1\right) \\ = & \Pr\left(v_t = 0\right) \\ = & \Pr(y \leqslant \xi) \\ = & F(\xi), \end{aligned} \tag{26}$$

so we get the expression $1 - F(y) \approx (F(\xi)) \left[1 - \frac{u_t}{\gamma}\right]^{\gamma}$ which is not in congruence with the main tail approximation in equation (19) as shown by Bi et al. (2023). Moreover, research by Chen et al. (2023) show similar weight derivations for the EVL loss function as it has been derived in equation (24). Therefore, applying the weights derived in equation (24) to the standard BCE loss function, we get:

$$\begin{aligned} \operatorname{EVL}\left(u_t, v_t\right) = & - \Pr(v_t = 1)\left[1 - \frac{u_t}{\gamma}\right]^{\gamma} v_t \log\left(u_t\right) \\ & - \Pr(v_t = 0)\left[1 - \frac{1 - u_t}{\gamma}\right]^{\gamma}\left(1 - v_t\right) \log\left(1 - u_t\right), \end{aligned} \tag{27}$$

where the standard BCE loss function for a binary classification task is given by:

$$\begin{aligned} \operatorname{BCE}\left(u_t, v_t\right) = & - v_t \log\left(u_t\right) \\ & - \left(1 - v_t\right) \log\left(1 - u_t\right). \end{aligned} \tag{28}$$

### 6.6 Additional Results

This section supports the findings in Section 4.2 with a series of qualitative results that provide a different perspective for assessing the quality of the results. Thus, the emphasis is on the quality difference between lead times on the extreme precipitation events dataset. Figure 3 aims to provide visualizations of the predicted lead times as a visual signal on the quality of the predictions. On the other hand, Figure 4 to Figure 16 provide additional analysis on the performance of the proposed models on the metrics presented in Section 4.2

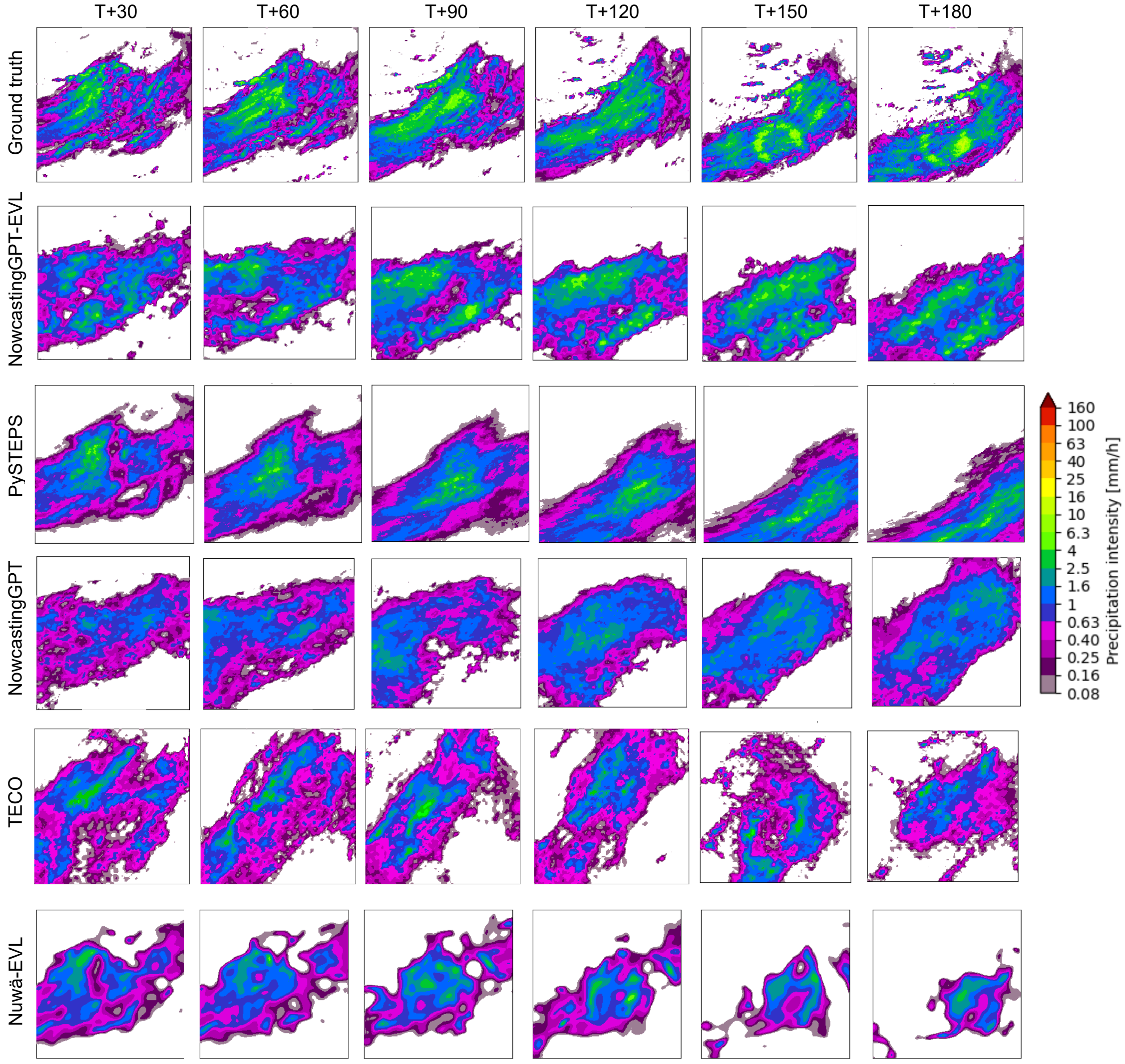

Figure 3: Nowcasting of extreme precipitation scenarios. The generation is conditioned on 3 previous timestamps with the task to predict the next 6 lead times. There is a gap of 30 minutes between each timestamp. Images are upsampled to 256 × 256 pixels.

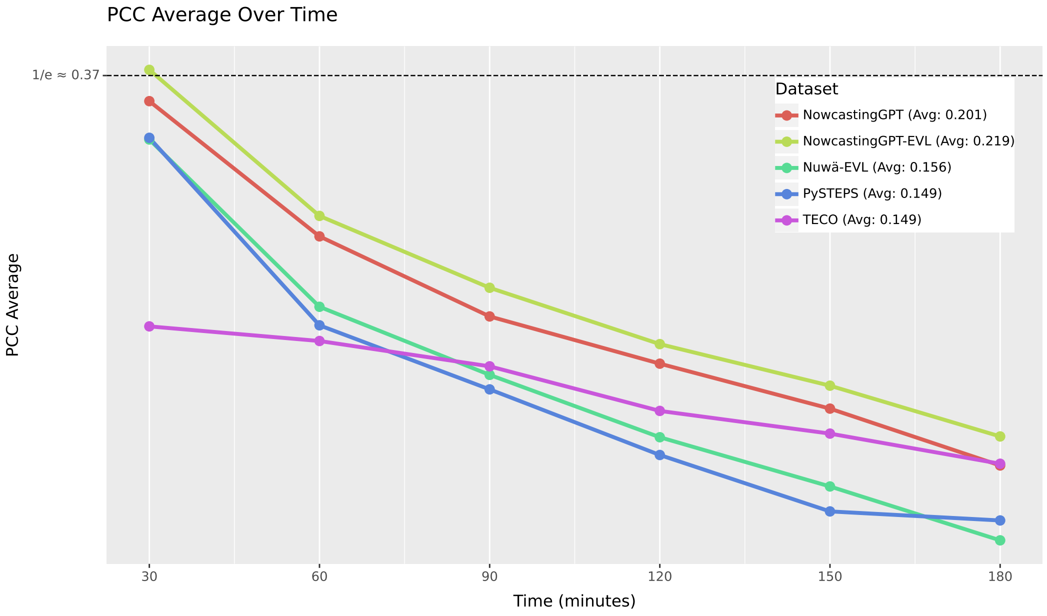

Figure 4: PCC metric evaluation over the 6 lead times. Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. NowcastingGPT-EVL outperforms all other models.

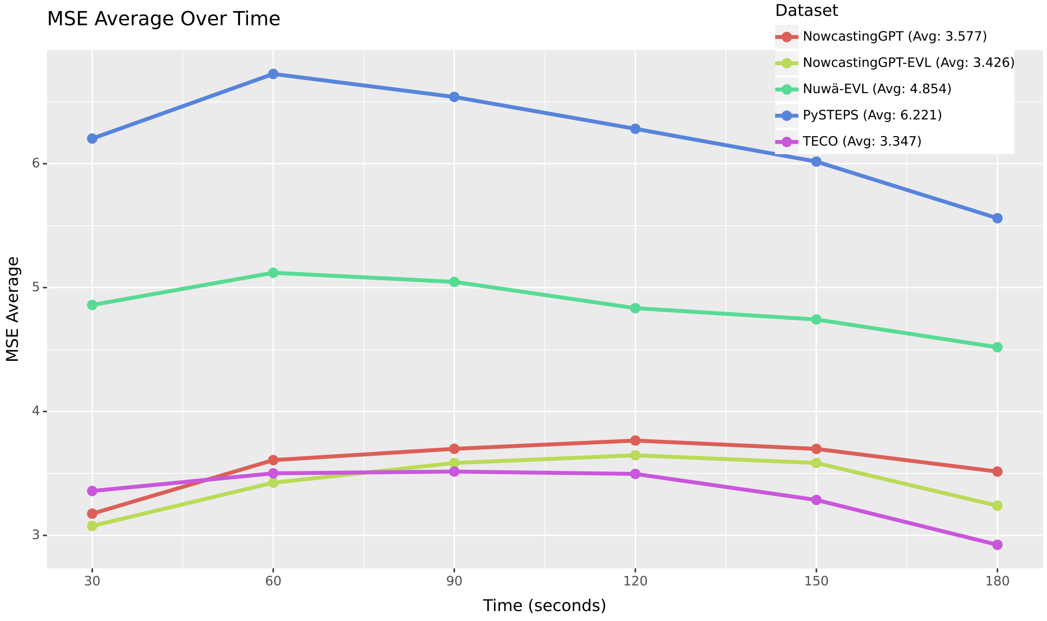

Figure 5: MSE metric evaluation over the 6 lead times. Each point represents the average value for a specific lead time over the whole dataset. Lower values represent better performance. NowcastingGPT-EVL and TECO outperforms all other models for bigger lead times.

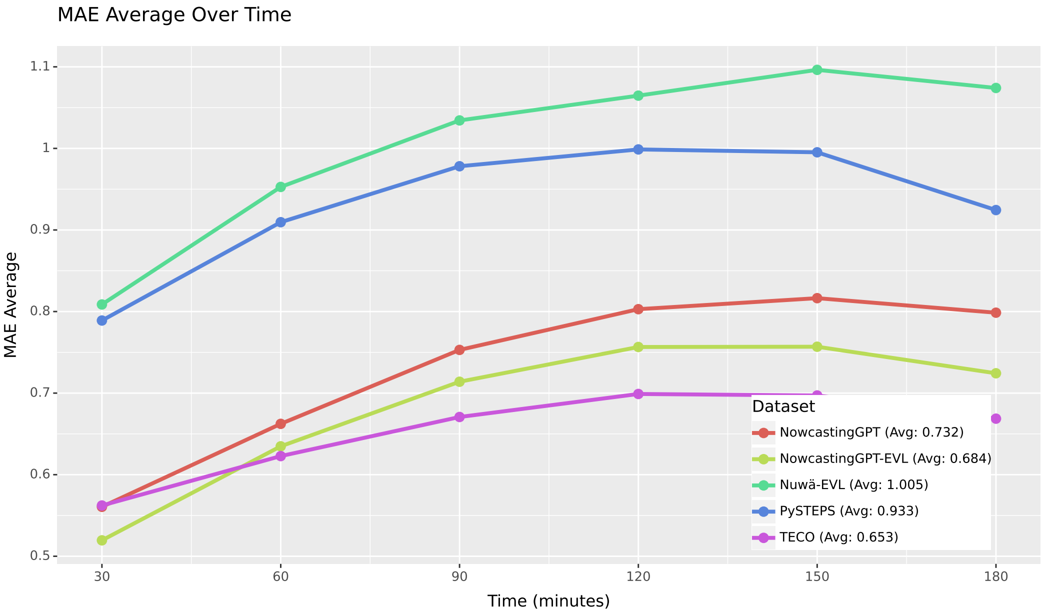

Figure 6: MAE metric evaluation over the 6 lead times. Each point represents the average value for a specific lead time over the whole dataset. Lower values represent better performance. TECO outperforms all other models.

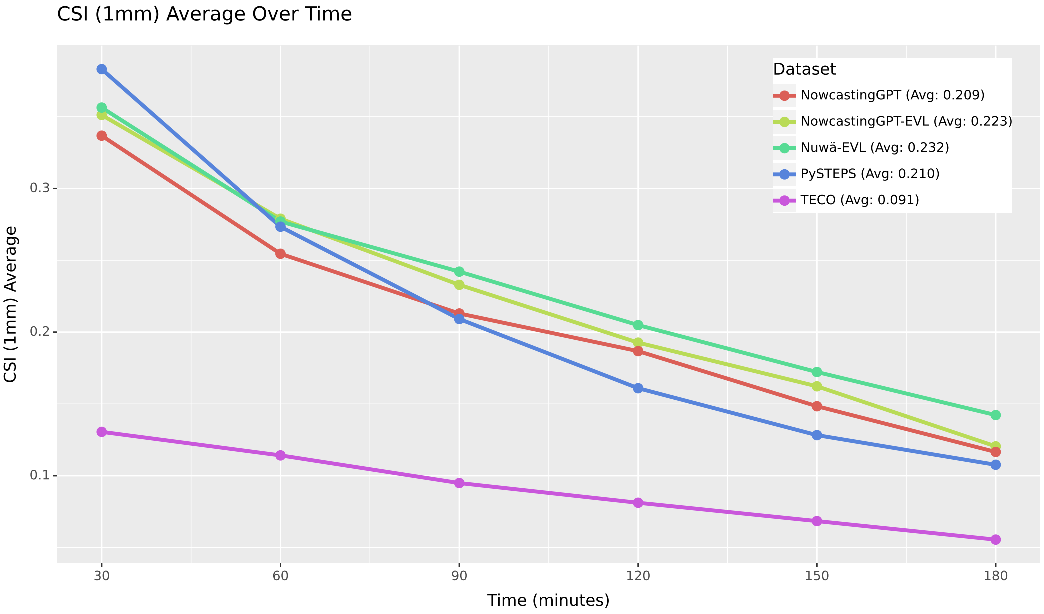

Figure 7: CSI(1mm) metric evaluation over the 6 lead times. Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. Nuwä-EVL and NowcastingGPT-EVL outperform the rest of the models but decay quickly.

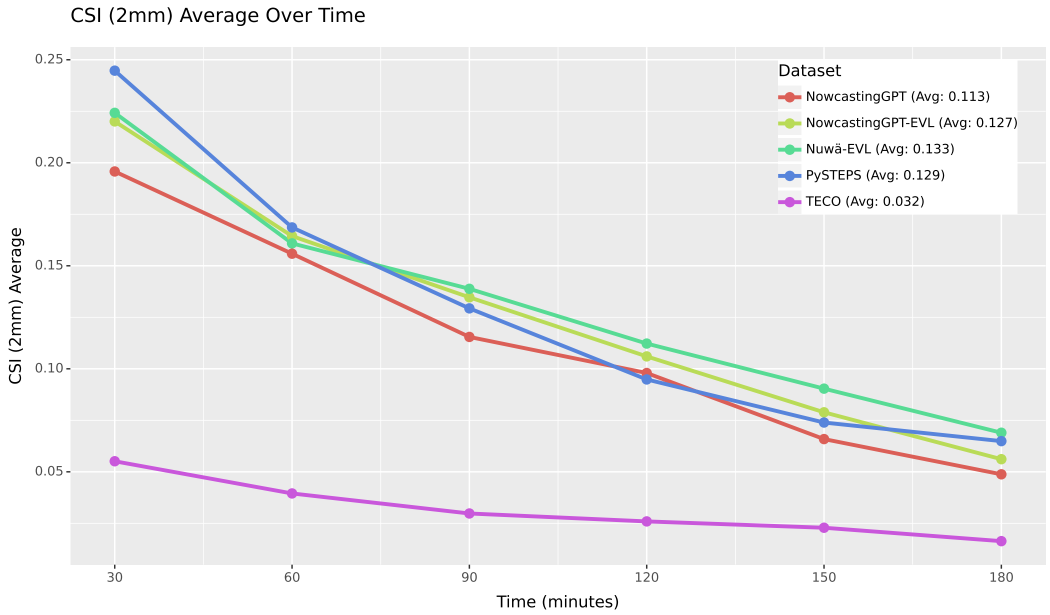

Figure 8: CSI(2mm) metric evaluation over the 6 lead times. Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. Nuwä-EVL outperforms the rest of the models.

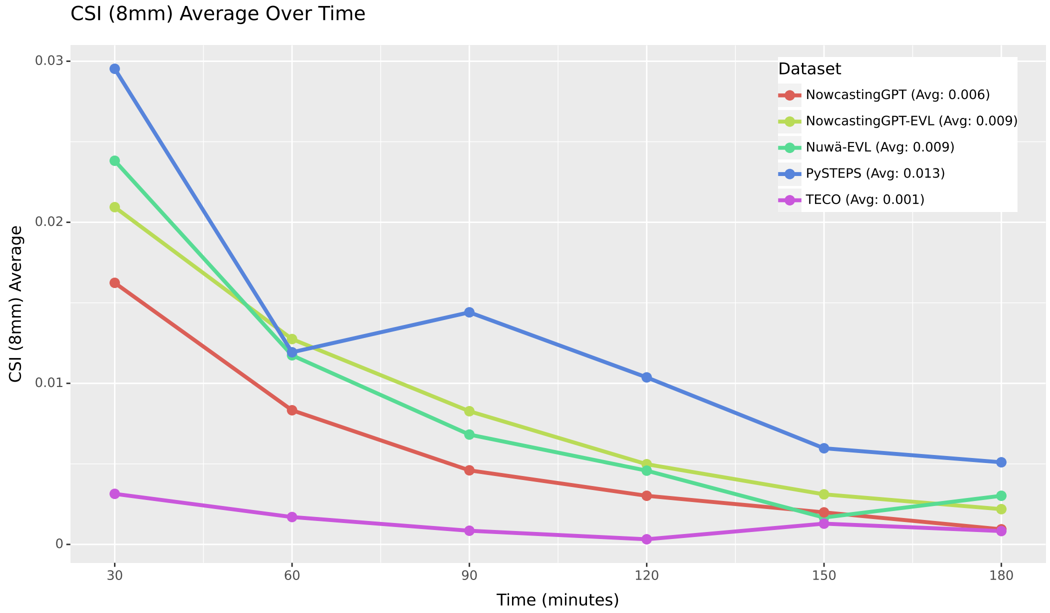

Figure 9: CSI(8mm) metric evaluation over the 6 lead times. Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. PySTEPS outperforms the rest of the models.

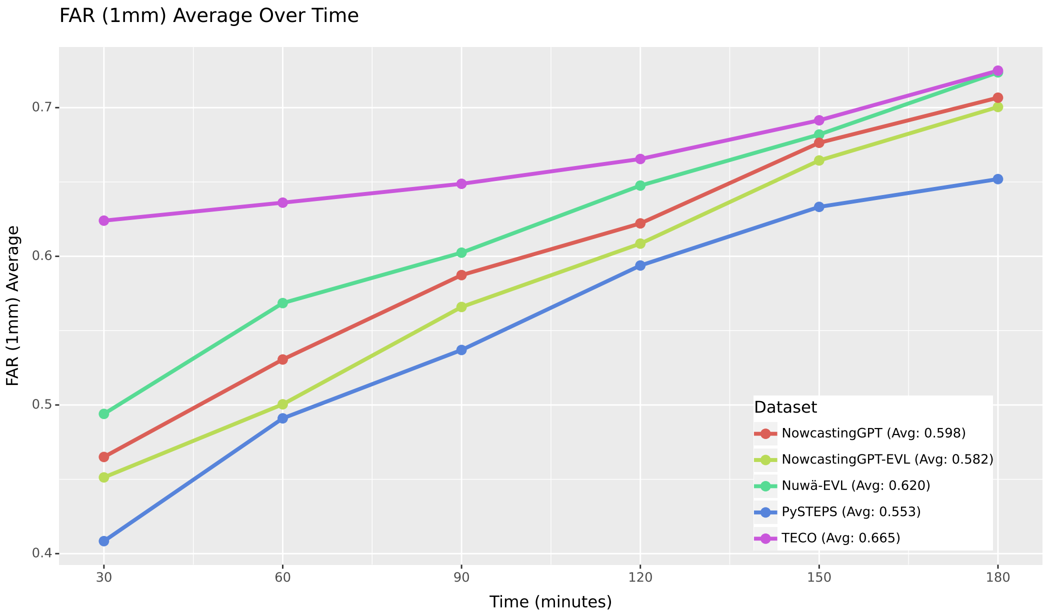

Figure 10: FAR(1mm) metric evaluation over the 6 lead times. Each point represents the average value for a specific lead time over the whole dataset. Lower values represent better performance. PySTEPS outperforms the rest of the models.

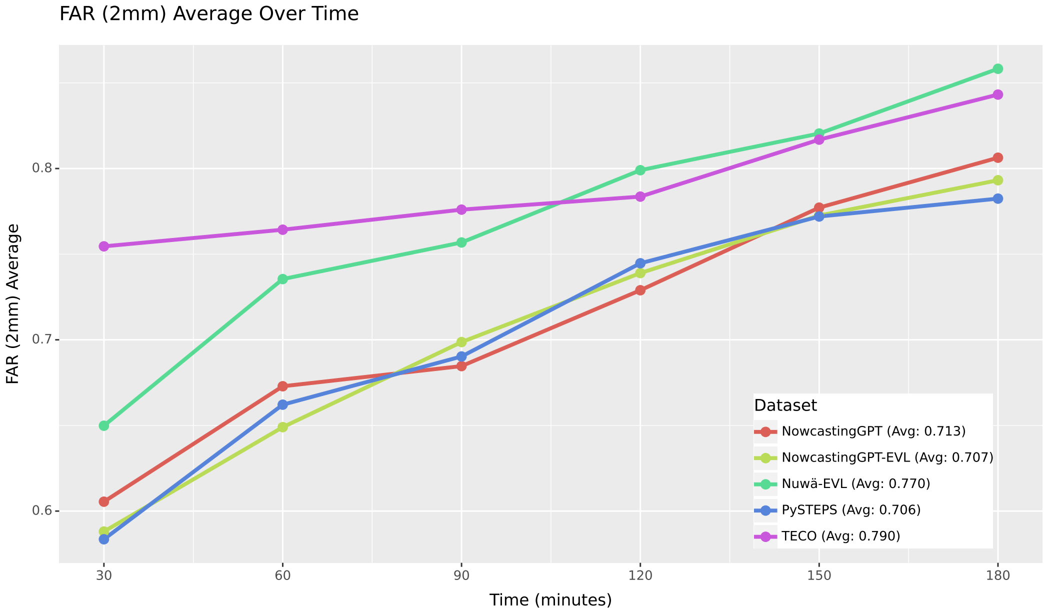

Figure 11: FAR(2mm) metric evaluation over the 6 lead times. Each point represents the average value for a specific lead time over the whole dataset. Lower values represent better performance. NowcastingGPT-EVL outperforms the rest of the models.

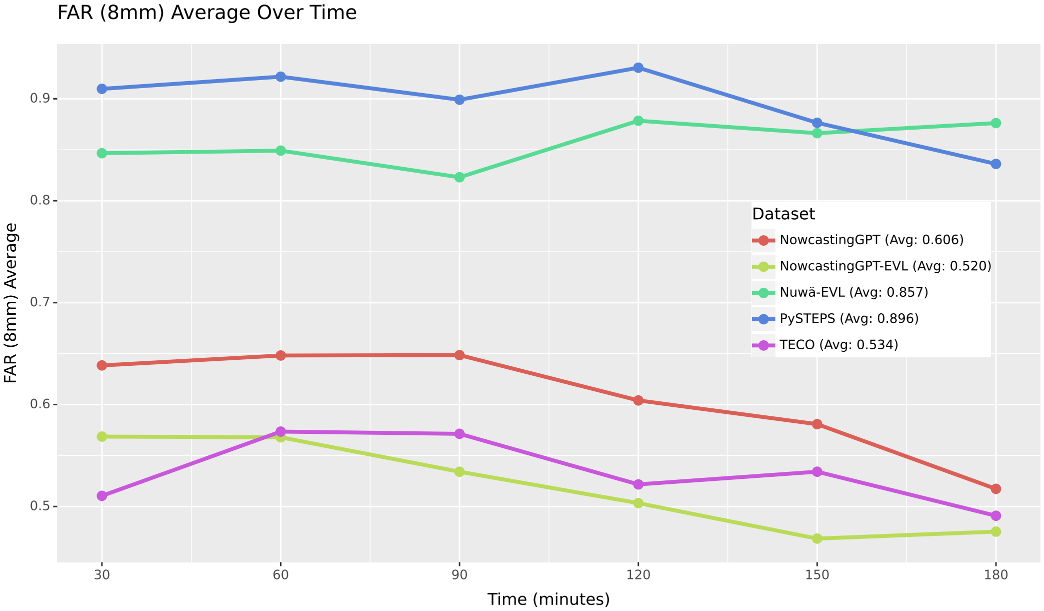

Figure 12: FAR(8mm) metric evaluation over the 6 lead times. Each point represents the average value for a specific lead time over the whole dataset. Lower values represent better performance. NowcastingGPT-EVL and TECO outperform the rest of the models.

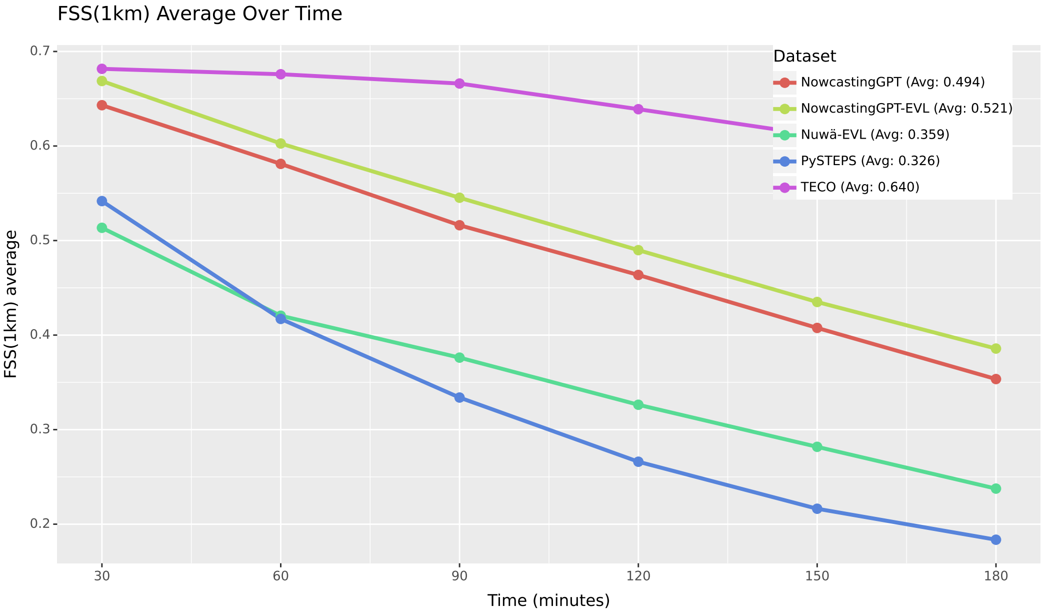

Figure 13: FSS(1km) metric evaluation over the 6 lead times. Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. TECO outperforms the rest of the models.

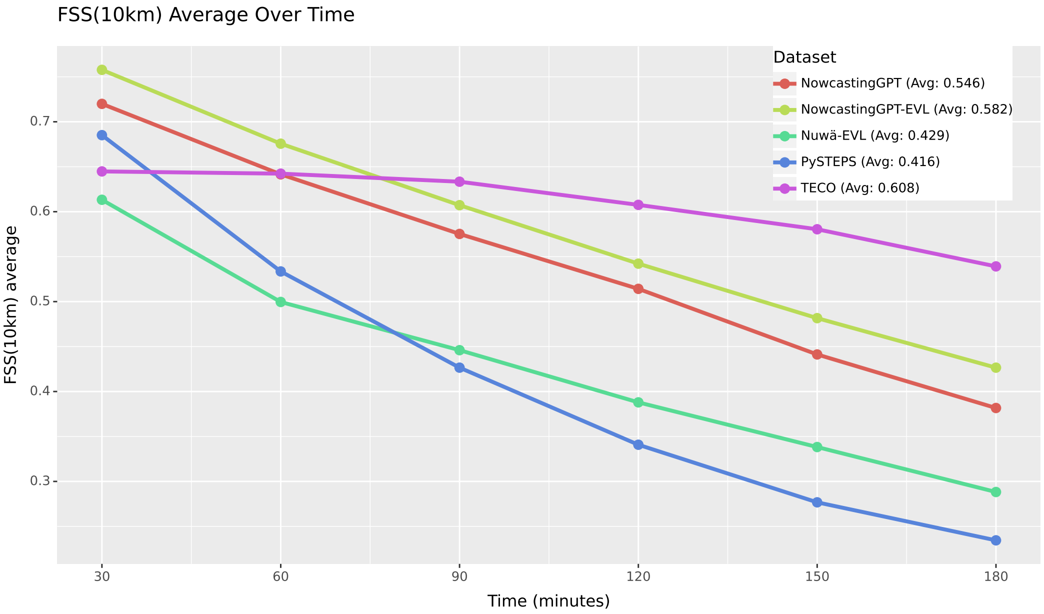

Figure 14: FSS(10km) metric evaluation over the 6 lead times. Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. TECO outperforms the rest of the models on higher lead times while NowcastingGPT and NowcastingGPT-EVL perform better on lower lead times.

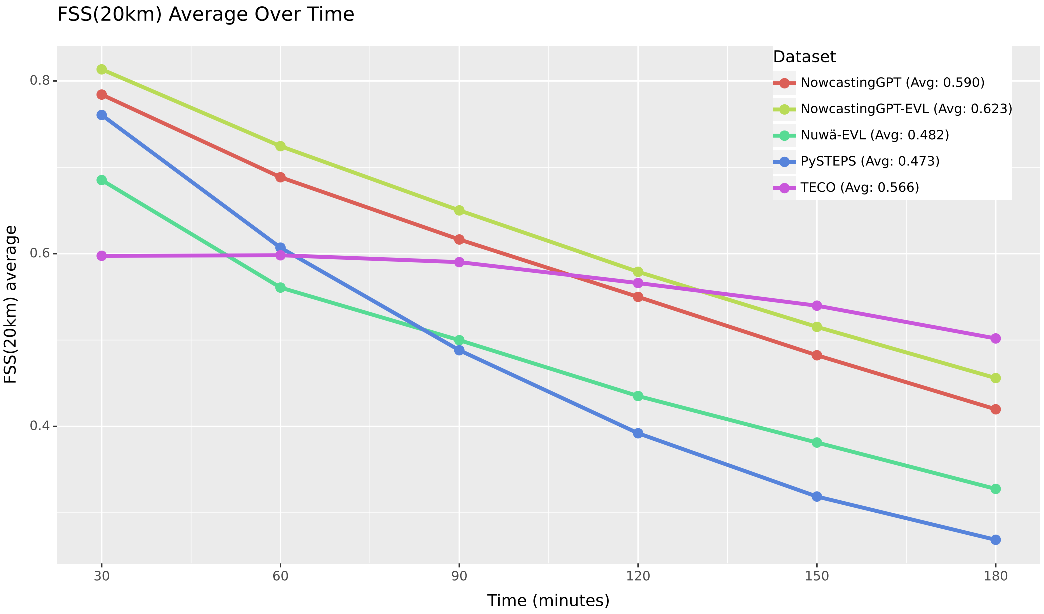

Figure 15: FSS(20km) metric evaluation over the 6 lead times. Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. NowcastingGPT-EVL outperforms the rest of the models.

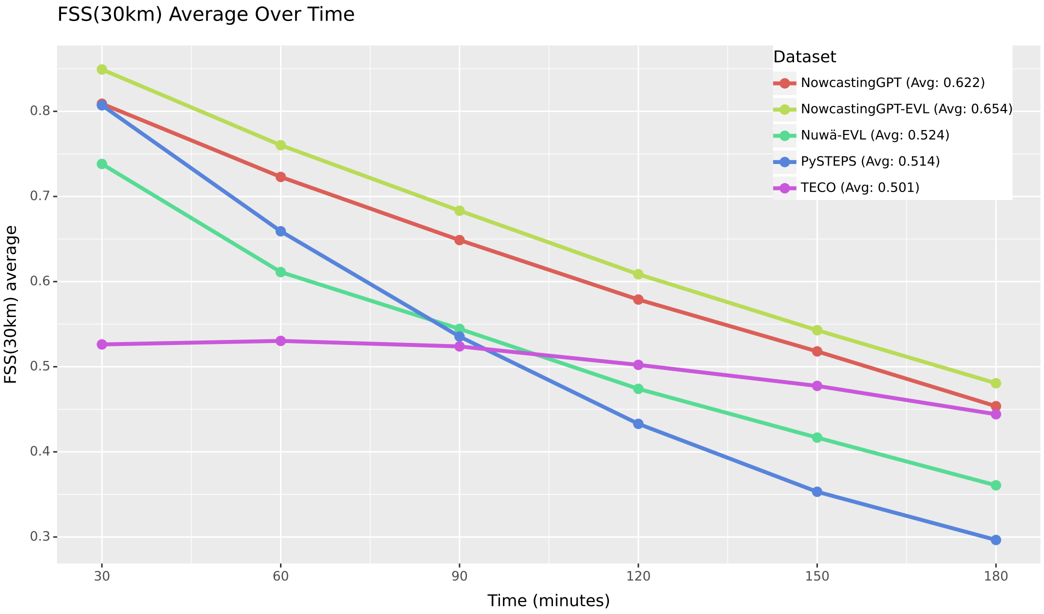

Figure 16: FSS(30km) metric evaluation over the 6 lead times. Each point represents the average value for a specific lead time over the whole dataset. Higher values represent better performance. NowcastingGPT-EVL outperforms the rest of the models.