Signal Processing Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
https://sps.ewi.tudelft.nl/

SPS-2024-00

M.Sc. Thesis

Extreme Precipitation Nowcasting using Transformer-based Generative Models

Ankush Roy B.Tech.

Abstract

Extreme precipitation, like floods and landslides, poses major risks to safety and the economy, underscoring the need for sophisticated weather forecasting to predict these events accurately, enhancing readiness and resilience. Nowcasting, which uses real-time atmospheric data to predict short-term weather, is key in addressing this challenge. Traditional nowcasting systems, reliant on extrapolation from rainfall radar observations and constrained by simplistic physical assumptions, often struggle to detect complex, nonlinear weather patterns. This gap has opened the door for deep learning models, which have shown significant promise in improving the accuracy and reliability of short-term weather predictions, making them a focal point of recent research and the basis of this thesis's approach.

This thesis introduces a deep generative model designed for the nowcasting of extreme precipitation events up to 3 hours ahead, utilizing a Vector-Quantized Variational Autoencoder (VQ-VAE) to compress radar data into a low-dimensional latent representation, and an Autoregressive Transformer for predicting future radar images. Additionally, a binary classifier works in conjunction with the Autoregressive Transformer to identify extreme versus non-extreme weather events, using these classifications to inform an Extreme Value Loss (EVL) function. This loss function aims to improve the accuracy of predicting extreme weather events by addressing the data imbalance between normal and extreme precipitation occurrences. The proposed model displays comparable performance with the state-of-the-art conventional methods and other deep learning nowcasting models in predicting extreme events.

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Extreme Precipitation Nowcasting using Transformer-based Generative Models

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Ankush Roy B.Tech.
born in Kolkata, India

This work was performed in:

Signal Processing Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Delft University of Technology
Department of
Microelectronics

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Extreme Precipitation Nowcasting using Transformer-based Generative Models "** by **Ankush Roy B.Tech.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 20-03-2024

Chairman: ______________________________
prof.dr.ir. Justin Dauwels

Advisor: ______________________________
prof.dr.ir. Justin Dauwels

Committee Members: ______________________________
prof.dr. Francesco Fioranelli

____

Abstract

Extreme precipitation, like floods and landslides, poses major risks to safety and the economy, underscoring the need for sophisticated weather forecasting to predict these events accurately, enhancing readiness and resilience. Nowcasting, which uses real-time atmospheric data to predict short-term weather, is key in addressing this challenge. Traditional nowcasting systems, reliant on extrapolation from rainfall radar observations and constrained by simplistic physical assumptions, often struggle to detect complex, nonlinear weather patterns. This gap has opened the door for deep learning models, which have shown significant promise in improving the accuracy and reliability of short-term weather predictions, making them a focal point of recent research and the basis of this thesis's approach.

This thesis introduces a deep generative model designed for the nowcasting of extreme precipitation events up to 3 hours ahead, utilizing a Vector-Quantized Variational Autoencoder (VQ-VAE) to compress radar data into a low-dimensional latent representation, and an Autoregressive Transformer for predicting future radar images. Additionally, a binary classifier works in conjunction with the Autoregressive Transformer to identify extreme versus non-extreme weather events, using these classifications to inform an Extreme Value Loss (EVL) function. This loss function aims to improve the accuracy of predicting extreme weather events by addressing the data imbalance between normal and extreme precipitation occurrences. The proposed model displays comparable performance with the state-of-the-art conventional methods and other deep learning nowcasting models in predicting extreme events.

Acknowledgments

First and foremost, I would like to sincerely thank Dr. Ir. Justin Dauwels Sir, my thesis advisor, for all of his invaluable guidance, support, and constructive feedback throughout the entire process of this research. His expertise and mentorship have been instrumental in shaping the direction of this thesis and refining its content. I am also thankful to my thesis supervisor, Cristian Meo for his insightful comments, suggestions, and encouragement. My heartfelt appreciation extends to the Signal Processing Systems group for providing the necessary resources and facilities. My deepest gratitude to my family for their unwavering encouragement, and understanding throughout this journey. Their unconditional support has been my pillar of strength, motivating me to persevere during challenging times. Special thanks to my peers Junzhe Yin and Zeina Bou Cher for their encouragement, and intellectual exchanges. Their friendship and support have made this academic endeavor a more enriching and enjoyable experience. In conclusion, I extend my sincere appreciation to all those who have contributed, directly or indirectly, to the completion of this thesis. Your support and encouragement have been invaluable, and I am truly grateful for your contributions.

Ankush Roy B.Tech.
Delft, The Netherlands
20-03-2024

Contents

List of Figures

List of Tables

Introduction 1

1.1 Background: Precipitation Nowcasting

Intense precipitation can significantly affect the economy, beginning with its impact on outdoor activities which leads to the postponement or cancellation of events. This disruption extends to delays in ground transportation, flight cancellations, interruptions or shutdowns of power stations, and the cessation of marine services. Furthermore, such precipitation can destroy agricultural crops, exacerbating the economic strain. As the consequences of high precipitation intensify, they evolve from mere inconveniences affecting both private and public events to posing serious threats to infrastructure, triggering landslides, compromising public safety, and, in the worst-case scenario, endangering human lives.

To mitigate these far-reaching impacts, the implementation of an early warning system becomes indispensable. Such a system would empower governments and responsible entities to take timely action to prevent such hazards. Weather Nowcasting acts as an early warning system, nowcasting is a method to predict the rainfall intensities over a specific region and at a short period of time (usually up to 6 hours).

This thesis concentrates on a weather prediction approach known as nowcasting. Nowcasting techniques are designed to forecast imminent weather changes over a brief period (usually under 6 hours), a timeframe during which Numerical Weather Prediction (NWP) models are less effective. NWP, the predominant method for predicting weather, utilizes mathematical representations of the atmospheric and oceanic conditions to forecast weather. However, initially, due to its intricate nature, NWP had a lower resolution and required significantly longer processing times than nowcasting methods, rendering it less appropriate for immediate forecasting needs [1]. Although advancements in computing have enhanced NWP's resolution in recent times, it still falls short of the predictive accuracy achieved by nowcasting systems. Despite nowcasting's limitation to forecasting only up to 1-6 hours ahead, its precise and dependable outcomes are crucial for promptly alerting against severe weather-induced risks like floods and landslides [2].

In precipitation nowcasting, the primary representation of weather conditions comes from radar precipitation fields generated by weather radars. These systems rely on a range of methods for extrapolating radar data. Typically, a nowcasting model's inputs consist of precipitation data from recent history (often spanning the last 1 to 3 hours), with the model's outputs being forecasts of upcoming radar precipitation fields.

Weather patterns are categorized into four levels of motion: global, synoptic, mesoscale, and microscale, in descending order of size. Initially, due to constrained computational capabilities, Numerical Weather Prediction (NWP) models were limited to low spatial and temporal resolutions. This limitation meant they could only identify patterns at the mesoscale and were unable to detect microscale phenomena, such as minor convective patterns critical for short-term rainfall forecasting. In response to this challenge, radar-extrapolation techniques were developed. These techniques utilize radar observations of the atmosphere's current state to predict future weather conditions. They offer the benefits of simpler models, higher resolution, and superior forecasting accuracy within the first few hours (typically less than 6 hours). Presently, radar-extrapolation approaches remain fundamental to the majority of operational nowcasting systems.

Weather patterns are categorized into four levels of motion: global, synoptic, mesoscale, and microscale, in descending order of size. Initially, due to constrained computational capabilities, Numerical Weather Prediction (NWP) models were limited to low spatial and temporal resolutions. This limitation meant they could only identify patterns at the mesoscale and were unable to detect microscale phenomena, such as minor convective patterns critical for short-term rainfall forecasting [1]. In response to this challenge, radar-extrapolation techniques were developed. These techniques utilize radar observations of the atmosphere's current state to predict future weather conditions. They offer the benefits of simpler models, higher resolution, and superior forecasting accuracy within the first few hours (typically less than 6 hours). Presently, radar-extrapolation approaches remain fundamental to the majority of operational nowcasting models.

In recent times, the integration of deep learning techniques for nowcasting applications has gained attention among researchers. Similar to traditional approaches, these advanced models predominantly utilize radar precipitation fields to depict weather conditions, aiming to forecast future precipitation patterns. The inaugural deep learning model for precipitation nowcasting, dubbed ConvLSTM, was introduced by Shi et al. in 2015 [3]. This innovative model approaches nowcasting as akin to predicting sequences in a video, employing convolutional operations within an LSTM framework to simultaneously capture spatial and temporal dynamics. Another research team approached the problem as one of image transformation, developing a deep learning framework grounded in the U-Net architecture [4]. While deep generative models like GANs and VAEs have achieved success in various deep learning domains, their application in precipitation nowcasting has also gained popularity in recent years. In 2021, DeepMind (Google) [5] constructed a conditional GAN model for nowcasting purposes (90 - 120 minutes prediction). Moreover, Bi et al. in 2023 [6], implemented a VQGAN model with an EVL loss function for the purpose of Extreme Precipitation nowcasting. The work of Bi et al. [6] has also been used as the benchmark for the work shown in this thesis as well.

Deep learning models for nowcasting present a stark contrast to traditional radar-extrapolation approaches by being entirely data-driven, highly adaptable, and not reliant on predefined physical laws. These models enhance the ability to capture non-linear phenomena through the incorporation of activation functions, addressing a key limitation of conventional methods in accurately simulating non-linear weather events, such as convection initiation. Despite their advantages, the application of deep learning to precipitation nowcasting poses unique challenges not encountered in other domains where deep learning has been successfully applied. Common issues include the generation of results with blurred features (especially, in longer horizons), and difficulties in modeling rare and extreme events. Further exploration and explanation of these nowcasting techniques are provided in Chapter 2.

1.2 Motivation

Traditional nowcasting techniques often rely on assumptions based on physical principles, which may not always be applicable in real-world scenarios, hindering their ability to detect crucial weather patterns. The advent of deep learning in the field of nowcasting offers a promising alternative. These advanced methods, driven purely by data, are capable of directly forecasting future precipitation without the need for physical assumptions. As a result, numerous deep learning-based nowcasting models have been developed. While these models generally demonstrate an enhanced ability to forecast low-intensity rainfall events more accurately than traditional approaches, their effectiveness diminishes significantly in predicting severe or heavy precipitation events [5]. The issue of blurry forecasts has been attributed by various scholars [5] to the absence of physical constraints and the reliance on mean square error (MSE) as a loss function. Furthermore, experts in the computer vision field have shown that adversarial training can yield sharper and more lifelike outputs. This concept was corroborated in a contemporary study on nowcasting by DeepMind [5], which introduced a GAN-based model specifically for precipitation nowcasting, achieving clear and precise predictions. Motivated by these promising results, our research intends to investigate the application of diverse deep generative models to enhance the nowcasting process.

Furthermore, a common issue with many of these models is their tendency to produce blurred and unrealistic forecasts for longer prediction intervals. However, Bi et al. [6] successfully addressed these challenges in their study, proposing a solution to the issues highlighted above. Consequently, the primary goal of this thesis is to build upon the foundational work of Bi et al.[6], aiming to develop a deep learning model that mitigates some of the constraints identified in their research.

The nowcasting challenge is distinct from conventional deep learning applications like video prediction, despite apparent similarities in inputs and outputs between the two. A key differentiator for nowcasting is its approach to managing extreme and anomalous events [1]. In standard tasks, such occurrences are often treated as outliers and excluded from consideration. Yet, in nowcasting, these rare but intense rainfall events are of paramount importance, as they can have significant economic and societal impacts. Most deep learning models for nowcasting fail to adequately account for these extreme precipitation patterns, leading to their omission in forecasts. Conversely, traditional techniques aimed at emphasizing these events, such as class weighting or oversampling, can introduce overfitting issues and result in exaggerated precipitation forecasts [7]. Given the challenges associated with accurately modeling both typical and extreme weather phenomena, this research investigates the application of extreme-value theory in developing a more effective model for nowcasting.

1.3 Research Goal

The objective of this study is to create a deep generative model focused on forecasting severe precipitation events within catchment regions in the Netherlands, offering forecasts up to 180 minutes ahead, at 30-minute intervals. This goal is divided into two key aims for the nowcasting system: firstly, to produce accurate and skillful forecasts across the entirety of the Netherlands; and secondly, to ensure that the model can consistently identify extreme precipitation events occurring within specific catchment areas.

To achieve the set goals, this thesis explores two primary subjects. The initial focus is on the creation of an innovative deep generative model for the nowcasting of precipitation. This model draws inspiration from recent advancements in visual synthesis technology, employing a dual-stage framework that integrates a VQVAE in its first phase and an autoregressive transformer in the subsequent phase. The second area of focus is the integration of extreme-value theory with the deep generative model to enhance the prediction of extreme weather events. This approach involves an addition of a binary classifier which classifies the tokens generated by the VQVAE as *extreme* or *non-extreme* and, adapting an extreme-value theory-derived loss function, termed Extreme Value Loss (EVL) [7]. The EVL loss function is integrated into the auto-regressive transformer component of our model.

Based upon this, the thesis work tries to cover two research objectives:

- To develop a deep generative model capable of reliably predicting precipitation fields for the upcoming 3 hours (180 minutes).
- To define and detect extreme precipitation events, and to accordingly adapt the model to enhance its ability to identify such extreme events.

1.4 Thesis Outline

In this section, the outline of the whole thesis has been defined:

1. Chapter 2 provides a comprehensive literature review of the existing nowcasting models. These models include both conventional numerical weather prediction models based on optical-flow (such as PySTEPS) as well as typical deep learning-based models. The subsequent sections of this chapter cover the significance of Extreme Value theory and its applications in modeling extreme behaviors in data.

2. Chapter 3 covers the details of the data used in this thesis work. At first, the KNMI radar dataset is introduced followed by the analysis of it in the later sections. Based on the statistics derived from the data analysis, specific problem statements are formulated. The chapter also covers the significance of extreme events specific to the scope of this thesis work.

3. Chapter 4 consists of the description of the proposed model in this thesis work. The model consists of a VQVAE with an auto-regressive transformer along with a binary classifier (similar to a vision transformer - to classify *Extreme* and *Non-Extreme* tokens). The auto-regressive transformer is mainly trained on the Cross entropy loss function but the transformer suffers from the huge imbalance in the dataset (very small number of extreme events). Therefore, the EVL (Extreme Value Loss) function is incorporated along with the cross entropy loss function (as regularization) to handle this class imbalance. The application of the EVL loss function is discussed in detail in this chapter. Finally, the different verification metrics for both the precipitation nowcasting task (in the whole Netherlands region) as well as extreme event detection task (on the catchment level) have been described in detail.

4. Chapter 5 presents the conducted experiments and the evaluation of the corresponding results. The experiments are mainly divided into two sections. The first section focuses on the evaluation of different nowcasting (both NWP and deep learning)

models on the whole Netherlands region whereas, the subsequent section describes the analysis of the extreme event detection on the catchment level. The detection performance is evaluated in two ways: one with defined and fixed extreme thresholds for different catchment areas, and the other with the same sets of extreme thresholds for catchments to assess the overall detection performance.

5. Chapter 6 encompasses the thesis's conclusion, providing a summary of the efforts and outcomes related to the proposed research objectives. Additionally, it outlines potential future avenues for this project, highlighting the limitations encountered during the thesis work and offering related recommendations and suggestions for future research.

Literature Survey 2

The literature review is divided into two main areas of interest: current approaches to nowcasting and the principles of extreme value theory. In the first part, we explore and compare traditional nowcasting techniques alongside more recent deep learning strategies. The second part delves into extreme value theory, a statistical field that examines the likelihood of rare events, presenting its basic concepts and how they are applied.

2.1 Existing Nowcasting Methods (Models)

In this section, the existing nowcasting models have been discussed in detail. The conventional nowcasting models cover topics such as Numerical Weather Prediction (NWP) models as well as radar-echo-extrapolation approaches whereas the subsequent section covers various types of deep learning models applied in nowcasting purposes.

2.1.1 Conventional Nowcasting Models

Traditional approaches to precipitation nowcasting models fall into two categories: those based on Numerical Weather Prediction (NWP) and those utilizing radar echo extrapolation. Due to the extensive computational demands of NWP techniques, radar echo extrapolation algorithms are more commonly employed in operational nowcasting systems, which is the primary emphasis of this discussion.

The radar-echo-extrapolation approach attempts to incorporate precipitation-related physics into simple methods, such as Euler persistence and Lagrangian persistence as mentioned by Germann et al.[8]. The Eulerian persistence approach utilizes the most recent observation as the predicted precipitation field. It can written as:

$$\hat{\Psi}\left(t_0+\tau, x\right)=\Psi\left(t_0, x\right), \tag{2.1}$$

where ψ is the observed precipitation field, t_0 is the initial time (start time of the forecast), τ is the lead time and, $\hat{\Psi}$ is the predicted precipitation field at time $t_0+\tau$. Eulerian persistence approach assumes that the precipitation fields remain static over time. Therefore, another approach is introduced in [8], known as the Lagrangian persistence method which takes into account the movement of the precipitation parcels. It is expressed as:

$$\hat{\Psi}\left(t_0+\tau, x\right)=\Psi\left(t_0, x-\lambda\right), \tag{2.2}$$

where λ is the displacement vector and the other variables have the same definition as mentioned above in equation (2.1). Most of the radar-based nowcasting methods are based on the Lagrangian persistence method.

Utilizing the premise that optical flow techniques from the field of computer vision can be adapted for nowcasting purposes, these methods typically involve two crucial steps: first, deducing the motion field based on observed data, and then, projecting the most recent observations forward along this motion field to create forecasts [1]. While Lagrangian persistence is a proven concept in the realm of precipitation nowcasting, serving as a cornerstone for numerous systems, its assumptions often fall short when predicting the actual movement of precipitation. To enhance the accuracy of nowcasting models, there's a shift towards incorporating probabilistic and stochastic methods. These methods not only account for the advection process but also introduce a measure of uncertainty into the forecasts. Essentially, they offer a more flexible approach to the concept of Lagrangian persistence by allowing for variations within the advection field [1].

The open-source initiative, PySTEPS, which encompasses a variety of the previously discussed precipitation nowcasting algorithms, serves as a benchmark for this project. The configuration and the workflow of PySTEPS has been described in the subsequent section.

Benchmark: PySTEPS

PySTEPS is a collaborative, open-source Python platform dedicated to precipitation nowcasting. It offers a range of algorithms for constructing nowcasting systems, supporting both deterministic and probabilistic setups [2]. This framework is widely recognized and regarded as leading-edge in the field of nowcasting tasks.

PySTEPS has both probabilistic as well as deterministic configurations which are guided by two main core algorithms namely, STEPS (Short-term Ensemble Prediction System) and S-PROG (Spectral PROGnosis). STEPS, a method for probabilistic forecasting, combines nowcasting outcomes with downscaled NWP (Numerical Weather Prediction) results [2]. The Royal Netherlands Meteorological Institute (KNMI) has recently adopted STEPS as its operational nowcasting system, and it will also serve as the benchmark for this thesis project. The implemented PySTEPS method in this thesis work follows the same configuration as mentioned in [2]. Specifically, the workflow of PySTEPS is as follows:

1. Read radar composites, transform the radar reflectivity data to rainfall (mm/h), then log-transform the result to dB scale.

2. Use the optical flow method to determine the motion field.

3. Use the advection method to extrapolate future radar precipitation field.

4. Use FFT to decompose the rainfall field into a multiplicative cascade, with each level representing a different spatial scale and rainfall intensity. An example of the decomposition result is shown in the below-mentioned figure 2.1.

5. Estimate the auto-correlation matrix for each cascade level, then estimate parameters for an AR model using Yule-Walker equations, and apply the model in time to handle temporal evolution and correlation within precipitation structure. The AR model is expressed as the equation below:

$$R_j(x, y, t) = \phi_{j,1} R_j(x, y, t - \Delta t) + \phi_{j,2} R_j(x, y, t - 2\Delta t), \tag{2.3}$$

   where R is the radar map, (x, y, t) is the coordinates, ϕ_j is the model parameter, and j is the number of cascade levels.

6. Add stochastic perturbations to the AR models and advection field. This way, the uncertainty in rainfall intensities and the motion field is considered.

7. Recompose the cascade with the AR model and the stochastic perturbations to get the result of the nowcasting ensemble.

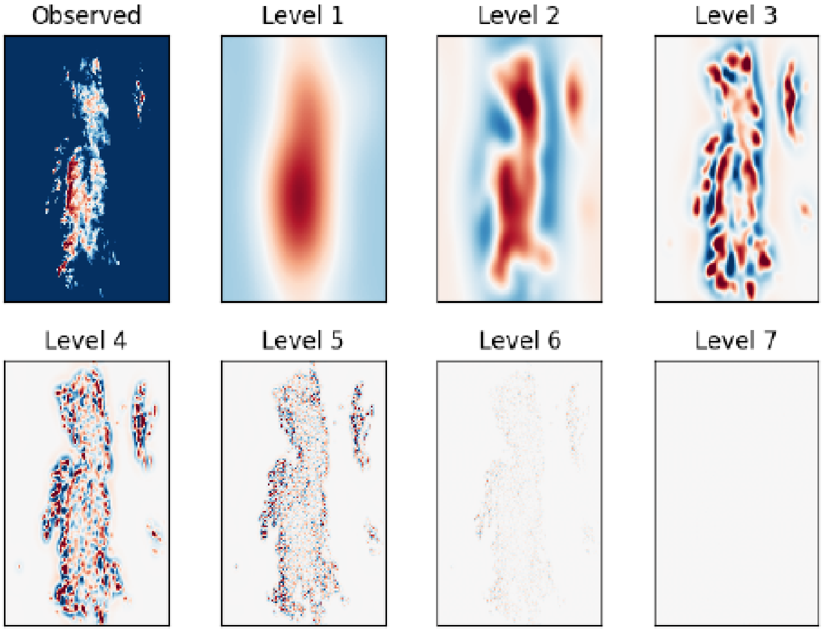


Figure 2.1: An example of the spatial decomposition at different cascade [9].

2.1.2 Deep Learning Models

Deep learning models surpass traditional reliance on mathematical representations of the atmosphere and meteorological assumptions, offering a more flexible approach. These models benefit from the large number of radar observation images, aiming to more accurately capture non-linear phenomena such as convective initiation and heavy precipitation. By directly predicting precipitation rates, they have demonstrated significant improvements, particularly at lower precipitation levels. The application of deep learning models in nowcasting can be divided into three perspectives (based upon the choice of the deep learning model) as: Spatial-temporal Convolution Networks, U-Nets, and deep generative models such as VQGAN, ClimaX (Vision Transformer based).

To achieve an efficient rainfall prediction, weather nowcasting should be treated as a spatial-temporal prediction task. Utilizing Long Short-Term Memory (LSTM) networks, a variant of Recurrent Neural Networks (RNNs), facilitates such spatiotemporal predictions. LSTMs have a unique gating mechanism that helps them remember and utilize information over extended periods, making them ideal for capturing the complex dependencies essential for accurate temporal and spatial weather predictions. The first such deep learning model used in precipitation nowcasting was introduced by Shi et al. in 2015 [3], known as the ConvLSTM. In ConvLSTM the traditional fully connected layers (in LSTM's state-to-state and input-to-state) are replaced with convolutional layers.

This enables the model to extract better spatio-temporal features, resulting in improved performance in nowcasting tasks when compared with the state-of-the-art conventional methods. Morevoer, additional improvements were made by Liu et al. [10] on the ConvLSTM model by incorporating the self-attention mechanism along with the memory module, known as ST-LSTM SA (Spatio-Temporal LSTM with Self Attention). The self-attention mechanism helps the model extract global and local dependencies between the extracted features and displays better nowcasting performance when compared with the traditional ConvLSTM.

Rather than viewing weather forecasting as a spatiotemporal prediction challenge, some researchers adopt an alternative perspective by framing nowcasting as an image-to-image translation issue. To address this, they utilize a renowned encoder-decoder architecture known as UNet. Unlike LSTMs, UNet does not have a specific component for memory modeling. It takes a radar image or a sequence of merged images as input and produces the subsequent forecast map as output. In simple terms, a U-Net is a CNN-based encoder-decoder architecture with a ”U” shape. The input to the U-Net are radar maps as images and it outputs a single map as the future frame [4]. Therefore, because of this reason, the prediction is carried out recursively in the case of U-Net.

Researchers have been refining the fundamental UNet architecture to enhance its predictive precision and efficiency. A significant innovation is the SmaAtUNet model, which integrates attention mechanisms and depth-wise separable convolutions [11]. By using Convolutional Block Attention Modules (CBAMs), the model systematically highlights salient features in both the channels and spatial dimensions of the input, leading to improved feature extraction. Despite having substantially fewer parameters, SmaAtUNet nearly matches the performance of the original UNet, offering a well-balanced trade-off between efficiency, speed, and accuracy.

Further advancements in model development have emerged from integrating attention mechanisms, as demonstrated by Bojesomo et al. with their model that combines 3D Swin Transformer blocks within a UNet structure [12]. This model is distinctive for its use of patch merging, multistage encoding with Swin Transformers, and a sliding window approach for localized self-attention, which allows it to capture a range of interactions. In the decoding phase, a cross-attention mechanism synergizes the encoder's outputs with the inputs, thereby significantly amplifying the model's ability to integrate and process information across various stages of the network.

In contrast to spatio-temporal networks, U-Net based prediction models enable tailored forecasting intervals, potentially enhancing short-term accuracy. They also benefit from a more simplified training approach. Nonetheless, it has been suggested that the iterative nature of this prediction method could result in error propagation over extended periods [13]. Many existing Deep Learning (DL) approaches focus on predicting weather for specific locations and at a specific temporal resolution, rather than providing probabilistic forecasts for entire precipitation fields. This limitation reduces their operational utility because they can't offer consistent predictions across multiple spatial and temporal scales.

Deep Generative Models are crafted to focus on probabilistic nowcasting. These models are data-driven and place a particular emphasis on the data's probability distribution. This emphasis enables them to encapsulate the inherent uncertainty in weather forecasting, thereby enhancing the accuracy and usefulness of the predictions. Ravuri et al. proposed Deep Generative Model for Radar (DGMR), the model uses a Conditional Generative Adversarial Network (GAN), where the generator is trained with losses from the two discriminators and a regularization term. The schematic diagram of DGMR is shown in the figure 2.2. DGMR is fed an input of four consecutive radar frames (with a temporal resolution of 5 minutes so the previous 20 minutes of radar observations) as context for the generator, allowing the prediction of 18 future precipitation maps (90 minutes lead time). The spatial discriminator aims to produce spatially consistent and detailed predictions while the temporal discriminator ensures temporal consistent predictions. This model gained improvement in terms of location accuracy, capturing and predicting small-scale weather phenomena, maintenance of statistical properties of precipitation, and avoidance of blurry predictions but has limitations in maintaining the intensity of heavy rainfall at longer lead times or extreme precipitation [5].

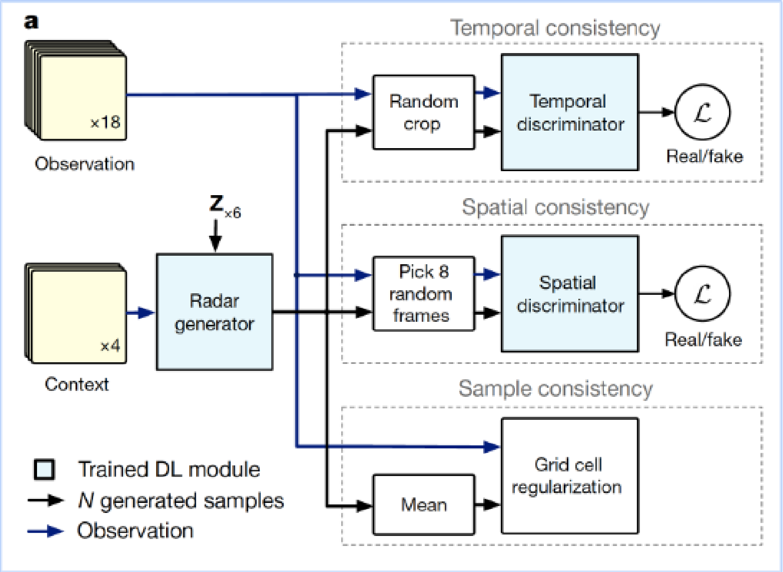


Figure 2.2: A schematic diagram of the model DGMR [5].

To solve the problem of predicting high-intensity rainfall, Bi et al. [6] proposed another deep generative model which consists of a Vector Quantization Generative Adversarial Network and a Transformer (VQGAN + transformer) known as, Nuwa-PyTorch (VQ-GAN). A schematic diagram of the model has been shown in figure 2.3. The functionality of the model is to convert the radar data into a compressed, efficient representation in the latent space using a Vector Quantized Variational Autoencoder (VQVAE) and a spatial discriminator to differentiate between the reconstructed and the original images. The auto-regressive transformer is able to predict future patterns using the latent space representation of the data from VQGAN, the attention mechanism used in the transformer is the 3DNA which is suitable for the 3D data structure. To handle extreme events, the Extreme value loss (EVL) is added to the cross entropy loss of the transformer. EVL is formulated on the basis of extreme value theory which helps the model to emphasize on better prediction of extreme events. Therefore, it can be concluded that the addition of EVL enhances the ability of the model to handle data imbalance by focusing on rare but extreme events [6].

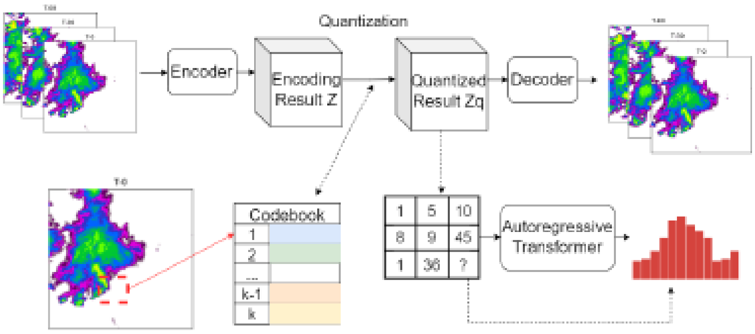


Figure 2.3: A schematic diagram of the model Nuwa-PyTorch (VQGAN) [6].

Nguyen et al. [14], introduced ClimaX, a flexible and generalizable deep learning model for predicting weather, which has been trained on a diverse array of datasets. This model builds upon the Transformer architecture, specifically the Vision Transformer, by incorporating vector tokenization and aggregation blocks. The research demonstrates ClimaX's flexibility and effectiveness for a range of applications, including local weather forecasting and long-term climate predictions. However, the model has not been applied to nowcasting, particularly for predicting severe rainfall events. Additionally, ClimaX requires a larger dataset for training compared to other deep generative models, which may limit its use in nowcasting extreme precipitation events due to the typically limited availability of such data.

Conclusion

Based on the above literature survey, it can be concluded that the key advantage of deep learning models is their flexibility when compared with conventional nowcasting methods. When comparing deep learning methods for nowcasting, models based on U-Net and ConvLSTM encounter two primary challenges. Firstly, while these models predict low-intensity rainfall with high accuracy, their performance significantly drops for high-intensity events, often leading to underestimations in the predicted radar maps. Secondly, the reliance on quadratic loss functions like mean square error (MSE) prompts the models to produce vague predictions as a way to mitigate the increased uncertainty associated with forecasts over longer periods.

The study of deep generative models for nowcasting is relatively new compared to other approaches. The Deep Generative Model for Radar (DGMR) demonstrates promising solutions to previously mentioned issues. It leverages adversarial training to enhance the sharpness of forecast images, addressing the problem of blurry outputs. Additionally, DGMR shows improved accuracy in forecasting medium to high rainfall intensities, marking a significant advancement in predictive capabilities for such weather events. Furthermore, ClimaX also shows promising results in the prediction of weather but the performance of ClimaX concerning extreme weather phenomenon is yet to be quantified. Meanwhile, Nuwa-PyTorch (VQGAN) also displays promising performance in the prediction of extreme precipitation when compared with the other models.

2.2 Extreme Value Theory and its applications

Extreme value theory is a segment of statistics that focuses on analyzing values that significantly diverge from the median in a probability distribution. This theory has found applications in predicting severe weather events [15], including instances of heavy rainfall and flooding. Furthermore, there has been exploration into its integration with deep learning models. For instance, the theory has been adapted into the loss function as seen in one study as the Extreme Value Loss (EVL) function [7], and in another, it has been combined with a Generative Adversarial Network (GAN) for the purpose of modeling spatial extremes [16]. While the theory encompasses both the analysis of minimum (left tail) and maximum (right tail) extreme values, the emphasis here is on the maximum values due to their relevance in forecasting precipitation events. The below-mentioned

sections contain the main definition of the Extreme Value Distribution function (including the different types of it), the applications of Extreme Value theory in various real-world scenarios as well as in deep learning task, and lastly, the mathematical derivation of the EVL loss function stated in [7] with relevant graphs describing the behavior of the loss function.

2.2.1 Generalised Extreme Value Distribution

Similar to how the Central Limit Theorem suggests that the sample mean will tend towards a normal distribution for large sample sizes from nearly any distribution, the Extreme Value Theory (EVT), also known as the Fisher-Tippett-Gnedenko Theorem, posits that for certain types of distributions, the maximum value in a large sample will conform to a Generalized Extreme Value (GEV) distribution [17]. Therefore, it can be said that Generalised Extreme Value Distributions are limiting distributions for the maxima of independent random variables sample from the same distribution. According to EVT, there are three distinct extreme value distribution models, commonly referred to as Gumbel, Fréchet, and Weibull distributions [18]. The cumulative distribution functions (CDF) for each are specified as follows:

- The Gumbel distribution (Type I) is characterized by a CDF that applies across all real numbers, indicative of data with lighter tails, similar to the normal distribution.
- The Fréchet distribution (Type II) describes heavier-tailed data distributions, typically used for modeling economic data and meteorological elements such as precipitation.
- The Weibull distribution (Type III) is appropriate for bounded variables, like certain environmental measurements.

These distributions merge into the Generalized Extreme Value Distribution (GEVD), represented by the following equation:

$$G(x) = \exp\left\{ - \left[1 + \xi \left(\frac{x - \mu}{\sigma} \right) \right]^{-1/\xi} \right\}, \tag{2.4}$$

Here, μ and σ denote the location and scale parameters, dictating the central tendency and variability of the distribution, respectively. The shape parameter ξ determines the form of the distribution: when $\xi = 0$, the model simulates the Type II distribution, and for $\xi > 0$ and $\xi < 0$, it corresponds to Type II and Type III distributions, respectively. Figure 2.4 shows the GEV distribution function with respect to different shape parameter values, thus, giving a better understanding of the influence of ξ on the shape of the probability density function of GEV.

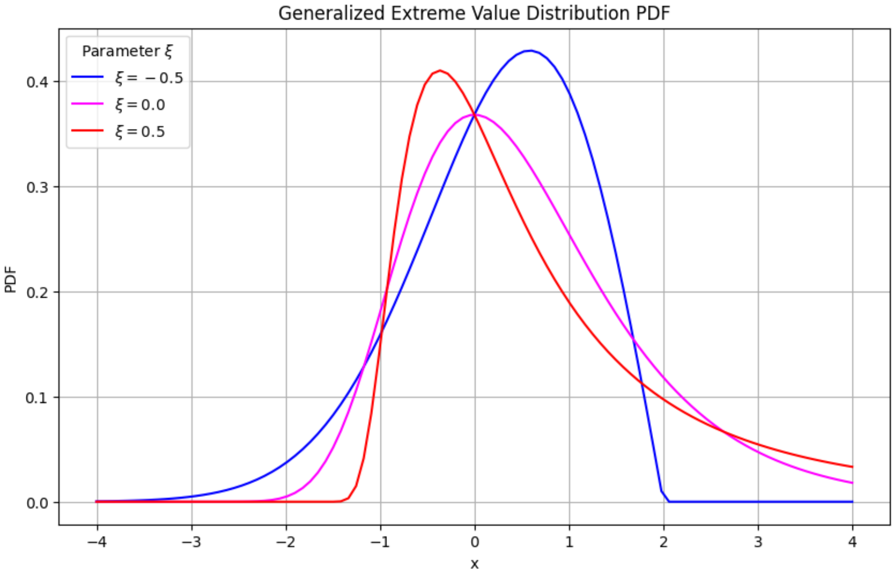


Figure 2.4: An example of GEV distribution with different shape parameters

The corresponding cumulative distribution function (CDF) versions for each type are as follows:

$$F(x) = \exp(-e^{-x}), \quad -\infty < x < \infty \tag{2.5}$$

$$F(x) = \begin{cases} 0 & x \leq 0, \\ \exp(-x^{-\alpha}) & x > 0, \alpha > 0 \end{cases} \tag{2.6}$$

$$F(x) = \begin{cases} \exp(-(-x)^{\alpha}) & x < 0, \alpha > 0, \\ 1 & x \geq 0 \end{cases} \tag{2.7}$$

2.2.2 Applications of Extreme Value theory

According to De Haan [18] and Coles [17], understanding the distribution of extreme events has several practical applications. For instance:

- Based on the annual highest tide measurements over the past 20 years, determine how high a barrier needs to be to avoid a once-in-a-century catastrophe where water overflows the barrier and floods the community.

- Other extreme events that can be studied: extreme temperature (cold or heat), wind speed (e.g. in a hurricane), water (drought or floods), earthquakes, forest fires, or financial collapse.

Research by Levine [19], highlights the significance of EVT in fields such as Risk management in financial markets. For instance, Omari et al. in their research [20] implement EVT to compute Value-at-Risk (VaR) forecast for a portfolio of currency exchange rates. Specifically, they use EVT for modeling the tail distribution of the daily stock market returns. Moreover, EVT has also been implemented in deep learning models in the recent years for time series prediction tasks. Boulaguiem et al. in 2022 [16], incorporated EVT with GANs (known as, evtGAN) to model the spatial dependencies between climate extreme variables such as precipitation and temperature. The process involves fitting extreme climate data to a generalized extreme value distribution (GEVD) for each location and then normalizing that data. Subsequently, a GAN is trained on the normalized data, generating new samples. These samples are then re-normalized to align with the original GEVD-fitted observations. The results of evtGAN have proven to be better than standard GANs as well as statistical approaches in spatial extremes tasks [16].

Ding et al. [7] in their research, identified that the limited capability of deep learning models to capture extremes is due to the traditional quadratic loss function. The authors suggest enhancing this by integrating extreme value theory into the loss function to improve detection of extreme events in time series analysis. The model, which is built upon a standard GRU framework, incorporates two novel elements: a memory network module that records and utilizes past extreme events for current predictions through an attention mechanism, and an Extreme Value loss (EVL) that employs weighted cross entropy to correctly classify extreme events, with weights derived from the extreme value theorem probabilities.

The Extreme Value Loss (EVL) function has been adopted in the studies by Bi et al. [6] and Chen et al. [21], demonstrating its utility in modeling extreme events. The following section will provide a mathematical derivation of the EVL loss function, illustrating its theoretical foundation and application in extreme event analysis.

2.2.3 Mathematical proof of the weights used in EVL loss function

As mentioned in [17], 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:

$$\Pr\{X > u + y \mid X > u\} = \frac{1 - F(u+y)}{1 - F(u)}, \quad y > 0. \tag{2.8}$$

Starting from the L.H.S we have :

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

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

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

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

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

According to [17], if the parent distribution F was known, then the distribution of threshold exceedances in equation (2.8) would also be known. However, that is not the case. Therefore, Coles et. al. [17], suggest 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 Random Variables (R.Vs) mentioned above (with common distribution function F), maximum order statistics has been used to characterize extremes:

$$M_n = \max\left\{X_1, X_2, X_3, \ldots X_n\right\}, \xrightarrow{P} x^*, n \to \infty. \tag{2.9}$$

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

$$P\left(\max\left(X_1, X_2, \ldots, X_n\right) \leqslant x\right) = Pr\left(X_1 \leqslant x, X_2 \leqslant x, X_3 \leqslant x, \ldots X_n \leqslant x\right). \tag{2.10}$$

Since, they are I.I.D we can also write equation (2.10) as,

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

Hence,

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

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 states that for a sequence of constants $a_n > 0$ and a real b_n there is a non-degenerate distribution function G stated as :

$$lt_{n\to\infty}\left[F\left(a_n x + b_n\right)\right]^n = G(x), \tag{2.11}$$

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

$$G(x) = \exp\left\{-\left[1 + \xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}, \tag{2.12}$$

where μ is the location parameter, σ is the scale parameter and, ξ is the shape parameter.

Also, equation (2.11) can be written as :

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

where G^* is another member of the GEV family. In [17], it is stated that if equation (2.11) allows the approximation of $\left[F\left(a_n x + b_n\right)\right]^n$ 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 equation (2.12) but with different values of μ, σ and ξ. Therefore, we can then write :

$$F^n(x) \approx \exp\left\{-\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}, \tag{2.13}$$

Taking natural logarithm on both sides,

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

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,

$$1-F(x) \approx \frac{1}{n}\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}. \tag{2.14}$$

Therefore, we substitute the above result obtained in equation (2.14) in the R.H.S of equation (2.8), for a large u and $y > 0$,

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

and,

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

Hence, equation (2.8) can be rewritten as:

$$\begin{aligned}\Pr\{X > u+y \mid X > u\} &\approx \frac{n^{-1}[1+\xi(u+y-\mu)/\sigma]^{-1/\xi}}{n^{-1}[1+\xi(u-\mu)/\sigma]^{-1/\xi}} \\ &= \left[1+\frac{\xi y/\sigma}{1+\xi(u-\mu)/\sigma}\right]^{-1/\xi} \\ &= \left[1+\frac{\xi y}{\tilde{\sigma}}\right]^{-1/\xi}.\end{aligned} \tag{2.15}$$

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 :

$$H(y) = 1-\left(1+\frac{\xi y}{\tilde{\sigma}}\right)^{-1/\xi}, \tag{2.16}$$

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

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

According to De Haan et al. [18], the above relation in equation (2.15) implies that, if block maxima have an approximating distribution G, then threshold excesses also 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 ξ. A rough mathematical relation between these two distribution functions can be derived as :

$$H(y) = 1 + \ln(G(y)). \tag{2.17}$$

for some location (μ) and shape ($\sigma, \tilde{\sigma}$) parameters.The relationship in equation (2.17) has also been utilized by Gencay et al. [22] to incorporate EVT in applications of Value-at-Risk for the relative performance of stock market returns in emerging markets. Consequently, equation (2.8) can be re-written with the help of the derived results in equations (2.15) and, (2.16) as :

$$\begin{aligned} \frac{1-F(u+y)}{1-F(u)} = \left[1+\frac{\xi y}{\tilde{\sigma}}\right]^{-1/\xi} \quad &\Longrightarrow \quad \frac{1-F(u+y)}{1-F(u)} = 1 - H(y) \\ &\Longrightarrow \quad 1-F(u+y) \approx (1-F(u))(1-H(y)) \end{aligned} \tag{2.18}$$

This is the main equation for the tail approximation of observations exceeding a threshold u [18] and can be written more formally as :

$$1 - F(x) \approx (1-F(t))\left\{1 - H_\xi\left(\frac{x-t}{f(t)}\right)\right\}, x > t \tag{2.19}$$

where H_ξ is the GPD function with the shape parameter ξ. Therefore, we use the result derived in equation (2.18) to derive the weights of the EVL loss function formulated by Ding et al. [7]. However, the authors of [7], utilize the GEV distribution function to define the underlying distribution of the time series data used in their experiment. The main objective of their experiment 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 I.I.D random variables $y_1, \cdots, y_T$ sampled from a distribution F_Y, the distribution of the maximum is realized using EVT as :

$$\lim_{T\to\infty} P\left\{\max\left(y_1, \cdots, y_T\right) \le y\right\} = \lim_{T\to\infty} F^T(y) = G(y), \tag{2.20}$$

for some linear transformation as mentioned in [7], where $G(y)$ is GEV distribution function. We can observe that equation (2.11) and equation 2.20 have the same meaning (but with different variables in their definitions). Moreover, the authors define the GEV function in the paper as :

$$G(y) = \begin{cases} \exp\left(-\left(1-\frac{1}{\gamma}y\right)^{\gamma}\right) & , \gamma \neq 0, 1-\frac{1}{\gamma}y > 0 \\ \exp\left(-e^{-y}\right) & , \gamma = 0 \end{cases} \tag{2.21}$$

where γ 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 (2.21) is similar to the definition mentioned in equation (2.12) but with $\xi = -\frac{1}{\gamma}$, $\mu = 0$ and, $\sigma = 1$. For modeling the tail distribution of the corresponding time-series data used in the research experiment presented in [7], the authors use equation (2.19). However, as mentioned before, rather than using the GPD function the authors use the GEV distribution function to model the tail approximation. Therefore, we substitute the relationship mentioned in equation (2.17) as $-\ln(G(y)) = 1 - H(y)$ in equation (2.19) and get the following result :

$$1 - F(y) \approx (1 - F(\xi)) \left[-\log G \left(\frac{y - \xi}{f(\xi)} \right) \right], y > \xi \tag{2.22}$$

where ξ is the threshold and, $f(\xi)$ is a scale function as mentioned in [7]. Also, the authors define an extreme indicator sequence $V_{1:T} = [v_1, \cdots, v_T]$ as :

$$v_t = \begin{cases} 1 & y_t > \xi \\ 0 & y_t \leqslant \xi \end{cases} \tag{2.23}$$

where ξ 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 mention a hard approximation for the term $(\frac{y-\xi}{f(\xi)})$ as u_t in equation (2.22) which is the predicted indicator by the neural network used in their research experiment. This can be interpreted as a normalization technique which restricts the values of the output y between $[-1, 1]$. Therefore, considering this to be true, equation (2.22) can be re-written as :

$$1 - F(y) \approx (1 - F(\xi)) \left[-\log G(u_t) \right], \tag{2.24}$$

Substituting the definition of GEV (as described in equation (2.21)) in the above equation (2.24) we obtain :

$$1 - F(y) \approx (1 - F(\xi)) \left[1 - \frac{u_t}{\gamma} \right]^{\gamma}, \tag{2.25}$$

The term $1 - F(\xi)$ (with the help of equation (2.23) and the definition of cumulative distribution function) can be written as :

$$1 - F(\xi) = \Pr(y > \xi) \implies 1 - F(\xi) = \Pr(v_t = 1), \tag{2.26}$$

where $\Pr(v_t = 1)$ is the proportion of extreme events in the dataset. Therefore, equation (2.25) with the above substitution, can be re-written as :

$$1 - F(y) \approx \Pr(v_t = 1) \left[1 - \frac{u_t}{\gamma} \right]^{\gamma}, \tag{2.27}$$

This tail approximation is incorporated as adaptive weights in the standard Binary Cross Entropy (BCE) loss function to define the main EVL loss function as mentioned in [7]. However, the authors in paper [7] define the weight as :

$$1 - F(y) \approx (1 - \Pr(v_t = 1)) \left[1 - \frac{u_t}{\gamma} \right]^{\gamma}, \tag{2.28}$$

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

$$\begin{aligned} & 1 - \Pr(v_t = 1) \\ =& \Pr(v_t = 0) \\ =& \Pr(y \leqslant \xi) \\ =& F(\xi). \end{aligned} \tag{2.29}$$

Therefore, we get the expression $1 - F(y) \approx (F(\xi)) \left[1 - \frac{u_t}{\gamma}\right]^{\gamma}$ which does not match with the main tail approximation in equation (2.22). However, Chen et al. [21] in their research have also utilised EVL loss function but the adaptive weights are in congruence with the weights derived in equation (2.27). Therefore, applying the weight derived in equation 2.27 to the standard BCE loss function, we get :

$$\begin{aligned} \mathrm{EVL}\left(u_t, v_t\right) = & -\Pr(v_t = 1)\left[1 - \frac{u_t}{\gamma}\right]^{\gamma} v_t \log\left(u_t\right) \\ & -\Pr(v_t = 0)\left[1 - \frac{1 - u_t}{\gamma}\right]^{\gamma}\left(1 - v_t\right)\log\left(1 - u_t\right). \end{aligned} \tag{2.30}$$

whereas the standard BCE loss function is given by :

$$\begin{aligned} \mathrm{BCE}\left(u_t, v_t\right) = & -v_t \log\left(u_t\right) \\ & -\left(1 - v_t\right)\log\left(1 - u_t\right). \end{aligned} \tag{2.31}$$

with u_t being the predicted probability and v_t being the binary label (0 or 1).

Dataset Analysis and Problem Formulation 3

3.1 Dataset

In this section, the details of the dataset used in this thesis work are explained. The section has been divided into two main categories namely:

1. The research concentrates on nowcasting within the Netherlands, utilizing precipitation data sourced from The Royal Netherlands Meteorological Institute (KNMI). This study examines the RT dataset which is the Real Time dataset obtained directly from the KNMI website. However, for event selection purposes another dataset is used alongside the RT known as, MFBS dataset.

2. Since, RT dataset is the primary dataset for the training of the deep learning model presented in this thesis work, a statistical analysis of the precipitation intensities has been presented in the second sub-section.

3.1.1 KNMI Radar Datasets

Weather radar is primarily utilized by meteorologists for observing precipitation. In the Netherlands, the Royal Netherlands Meteorological Institute (KNMI) operates two C-band weather radars situated in Den Helder and Herwijnen. Prior to 2017, the operational radar was located at De Bilt, before being replaced by the one in Herwijnen [23]. The locations of these KNMI weather radars, along with a real-time radar map, are depicted in the below-mentioned Figure 3.1. Weather radars generate initial data known as radar reflectivity Z, representing the quantity of radiation reflected back at an altitude of 1500m across the Netherlands, as relevant to the datasets discussed in this thesis.

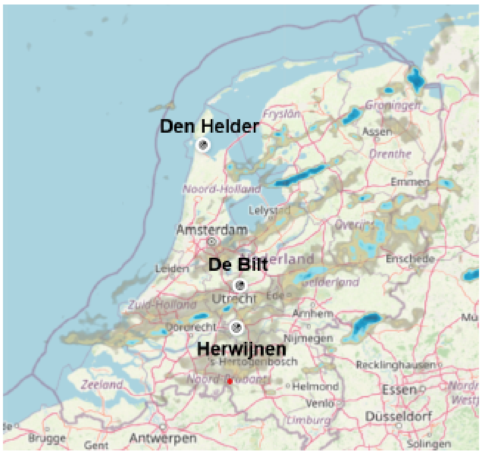


Figure 3.1: Three Radar stations in the Netherlands: Den Halder, Herwijnen and De Bilt [23].

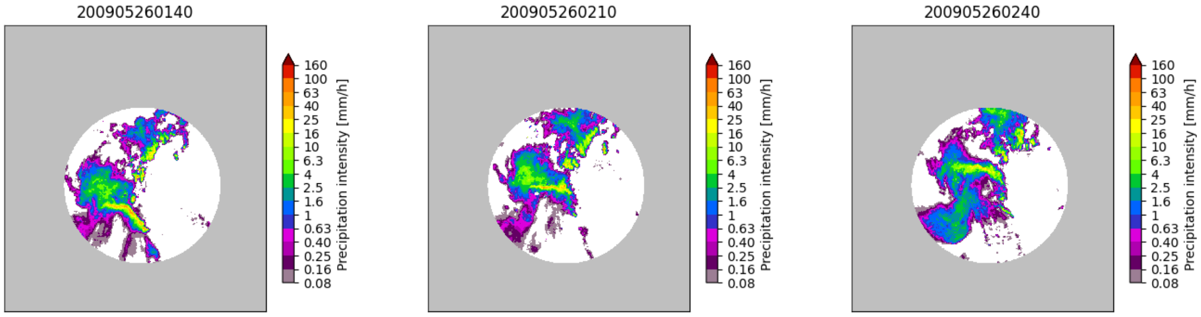
For the estimation of the rainfall rate from the radar reflectivity, a fixed Z-R transformation is implemented [24] given by:

$$Z_h = 200R^{1.6}, \tag{3.1}$$

where Z_h represents the radar reflectivity (unit: mm^6m^{-3}) and R represents the rainfall rate (unit: $mm\,hr^{-1}$). The original radar reflectivity has the unit of dBZ which can be converted to mm^6m^{-3} using $Z_{h(dBZ)} = 10\log_{10}(Z_h)$. In this process, reflectivity below 7dB (precipitation intensity $< 0.1mm\,hr^{-1}$) are ignored and reflectivity above 55dB (precipitation intensity $> 100mm\,hr^{-1}$) are fixed at 55dB. Moreover, isolated pixels have been ignored. This discard of reflectivity values is mainly done to prevent noise (precipitation intensity $< 0.1mm\,hr^{-1}$) and also, to prevent strong residual clutter-induced reflection (precipitation intensity $> 100mm\,hr^{-1}$) [23]. Therefore, the data after the aforementioned conversion is the Quantitative Precipitation Estimation (QPE) data from the KNMI website, known as the RT dataset [25].

In addition to the previously mentioned RT dataset, the KNMI website [26] also offers access to the MFBS dataset. This dataset shares the RT dataset's spatial and temporal resolutions. Nevertheless, it is refined using data from a comprehensive network of rain gauges, which includes 356 gauges spread across the Netherlands, to fine-tune the initial QPE. Consequently, it yields a more precise estimation of the rainfall rate. Despite its accuracy, the MFBS dataset is updated on a monthly basis on the KNMI website, and the extensive manual labor involved in operating the rain gauge network means that the dataset does not provide real-time (RT) data (it is updated once a day). Therefore, the RT dataset has been chosen as the main dataset for training and testing the deep learning model proposed in this thesis work. However, for the selection of the rainy events as mentioned in section 3.3, the MFBS dataset has been utilised since, it contain better estimation of the rainfall intensities.

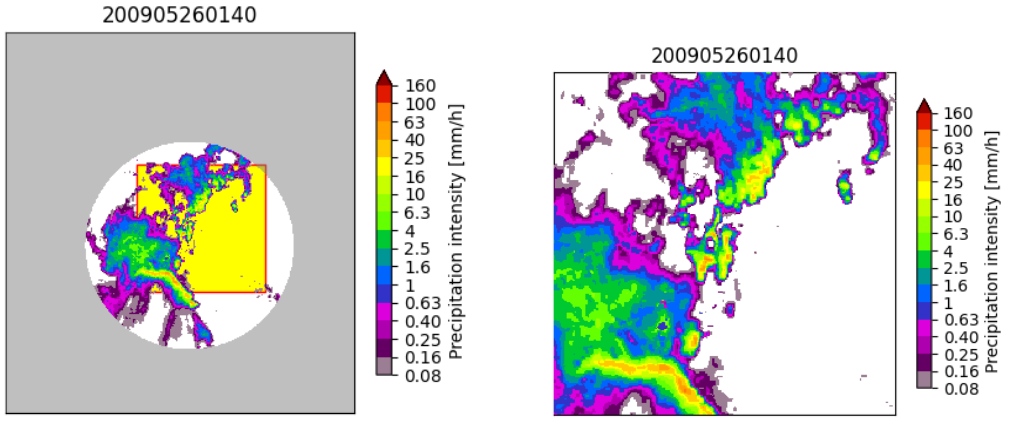
The RT dataset contains radar images from the year 2008-2021. Each radar map is 765 × 700 image with a spatial resolution of 1km and, a temporal resolution of 5 minutes. This means that the dataset contains radar images at every 5 minutes for the whole region of Netherlands.



(a) (b) (c)

Figure 3.2: Example of Radar Images from the RT dataset for three different time stamps.

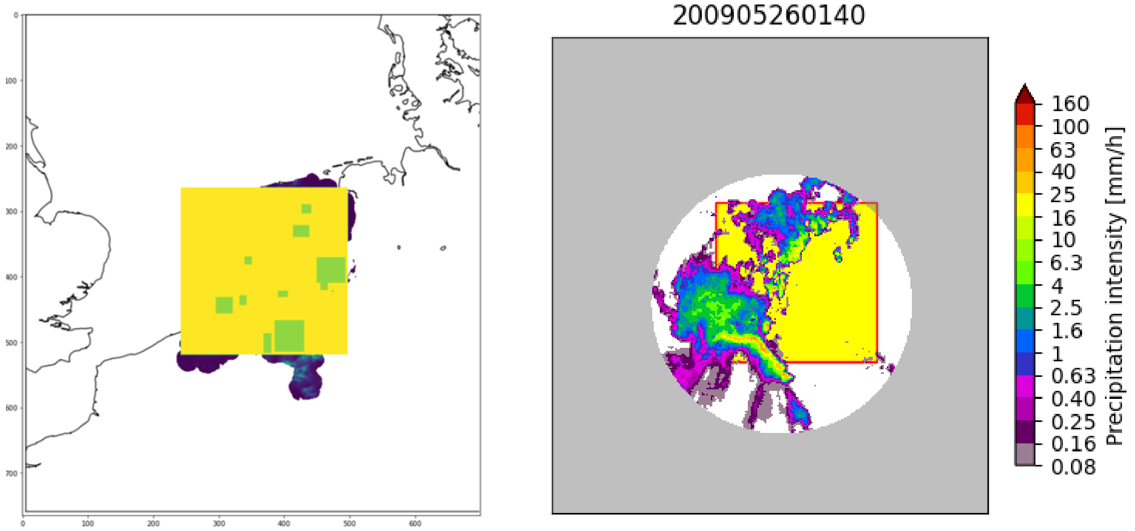
An example of the images from the RT dataset has been shown in the above-mentioned figure 3.2 from the year 2009, and the month May. From the figure, it can be observed that most of the area in the images is masked (the circular region has a diameter of approximately 400km). The circular area covers the Netherlands region along with some surrounding areas from Belgium and Germany. However, the main objective of this thesis is to perform extreme precipitation nowcasting with regards to the region of Netherlands. Therefore, we crop the respective images in the RT dataset with dimensions 256 × 256 as shown in the below-mentioned figure 3.3.



(a) **(b)**

Figure 3.3: Example of Radar Image from the RT dataset with the area of $256km \times 256km$.

The main reason behind this cropping is that it covers most of the land area of the Netherlands including all the 12 catchments where it is crucial to identify the occurrences of extreme precipitation. In figure 3.4 presented below, it can be observed that the *red* bounding box covers all the 12 catchments, part of this study. The left sub-figure 3.4a, shows the actual locations of the catchments in the Netherlands map while in the right sub-figure 3.4b the catchment regions are highlighted with respect to the RT dataset images.



(a) **(b)**

Figure 3.4: Representation of the catchment regions in the $256km \times 256km$ area chosen for this thesis project.

3.1.2 Data Analysis

In this section, the analysis of the RT dataset images has been done. Firstly, the precipitation distribution of the pixels in the whole map (entire Netherlands region) has been analysed followed by the analysis in the 12 catchments selected for the detection of extreme precipitation.

For the analysis in the whole Netherlands region, a total of 60,000 radar images spanning from the years 2008 to 2014 were chosen (the main reason behind the choice of this interval of year is that it constitutes the training dataset). The first image is taken from **00:00, 01/01/2008**, and then one image sampled every hour consecutively (60,000 times). Table 3.1 shows the occurrence of different precipitation intensities in the respective images mentioned above.

Rainfall intensity X	**Occurrence**	**Percent**
$X \leq 0.1\,mm/h$	3.724E+09	94.7%
$0.1\,mm/h < X \leq 1\,mm/h$	1.653E+08	4.2%
$1\,mm/h < X \leq 5\,mm/h$	3.992E+06	1.0%
$5\,mm/h < X \leq 10\,mm/h$	2.356E+06	0.06%
$10\,mm/h < X \leq 20\,mm/h$	5.213E+05	0.013%
$X > 20\,mm/h$	1.731E+05	0.004%

Table 3.1: Summary of the occurrence of different types of rainfall intensities in pixels for the whole Netherlands region.

Moreover, based upon the data in table 3.1, a histogram along with an empirical Cumulative Distribution Function (CDF) plot of the precipitation intensities for all these images have also been constructed and shown in figures 3.5 and 3.6.

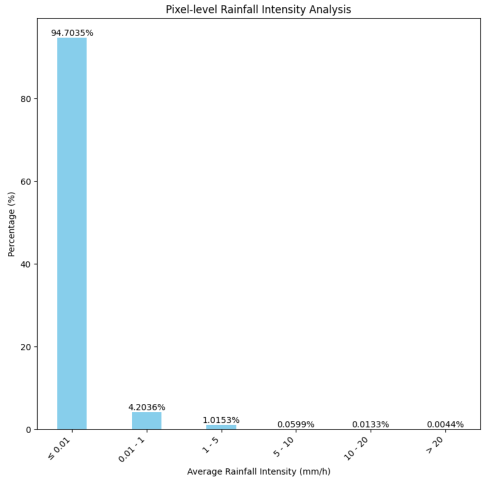
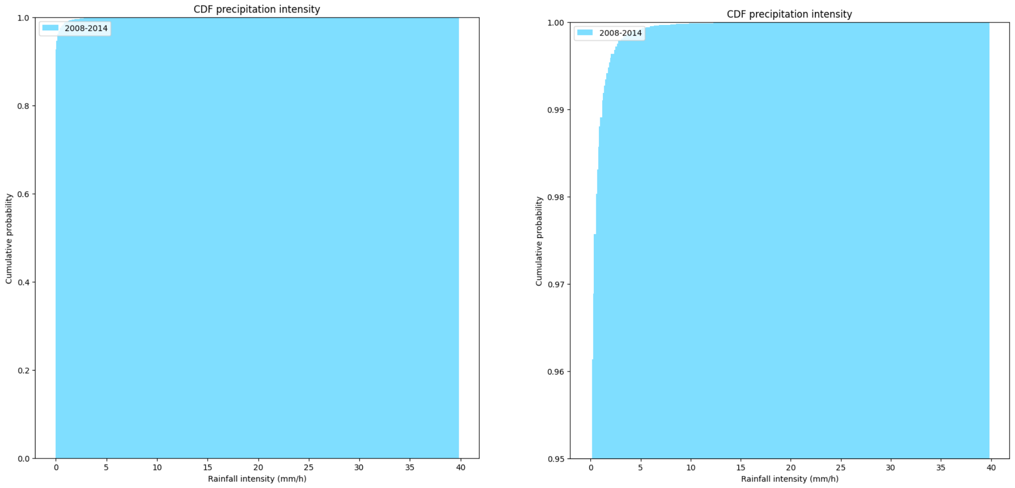


Figure 3.5: Histogram of the precipitation intensities distribution of the selected RT images.

It can be observed from figure 3.6 that lower rainfall ($X \leq 1\,mm/h$) intensities are much more common than higher ($1\,mm/h < X \leq 5\,mm/h$) and very high ($X > 10\,mm/h$) intensities. From sub-figure 3.6b, it can be seen that the Cumulative Probability rises quite steeply till 0.98 (approximately) suggesting that the images contain low-intensity rainfall the most. As the rainfall intensity increases, the CDF curve becomes less steep and starts to plateau, indicating that higher rainfall intensities are less common. For example, the cumulative probability doesn't increase much beyond a certain point on the x-axis (10mm/hr approximately), suggesting that very high rainfall intensities are rare.



(a) **(b)**

Figure 3.6: Representation of the Cumulative Distribution of the different rainfall intensities.

Catchment regions are defined as territories where surface runoff gets collected in particular locations. These areas are the most susceptible to floods and stagnation of water after heavy rainfall because of their locations. Therefore, the nowcasting outcomes for these catchment regions are of significant importance because they can be employed in hydrological models to provide early warnings of potential flooding. In the scope of this thesis work, 12 Dutch catchments will be analyzed as shown in figure 3.4a. A detailed locations of these 12 catchments along with their respective area coverage have also been shown in the below figure 3.7 and, table 3.2.

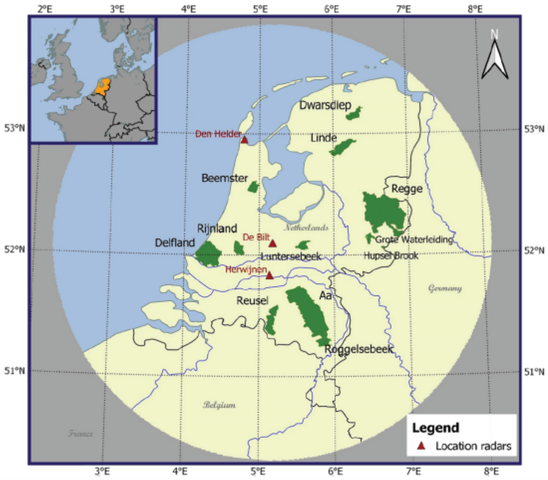


Figure 3.7: Map of the Netherlands [6].

Number	Catchment name	Area (km^2)
1	Regge	957
2	Aa	836
3	Delfland	379
4	Reusel	176
5	Linde	150
6	Rijnland	89
7	Roggelsebeek	88
8	Dwarsdiep	83
9	Beemster	71
10	Luntersebeek	63
11	Grote Waterleiding	40
12	Hupsel Brook	6.5

Table 3.2: Catchment areas and their respective sizes.

For the analysis in the catchment level, the average rainfall accumulation has been calculated for a 3-hour window for all the relevant catchment areas. This serves as the main indicator of the rainfall intensity for the catchment areas. The radar images selected for analysis correspond to the same period, namely 2008 to 2014, which was used for the comprehensive analysis of the Netherlands region. Figure 3.8 shows the distribution of the 3-hour average rainfall intensity. It can be observed that approximately 90 % of the catchment averaged precipitation is smaller than 1mm/3hr (similar to the analysis shown in figure 3.5, for the whole region of the Netherlands).

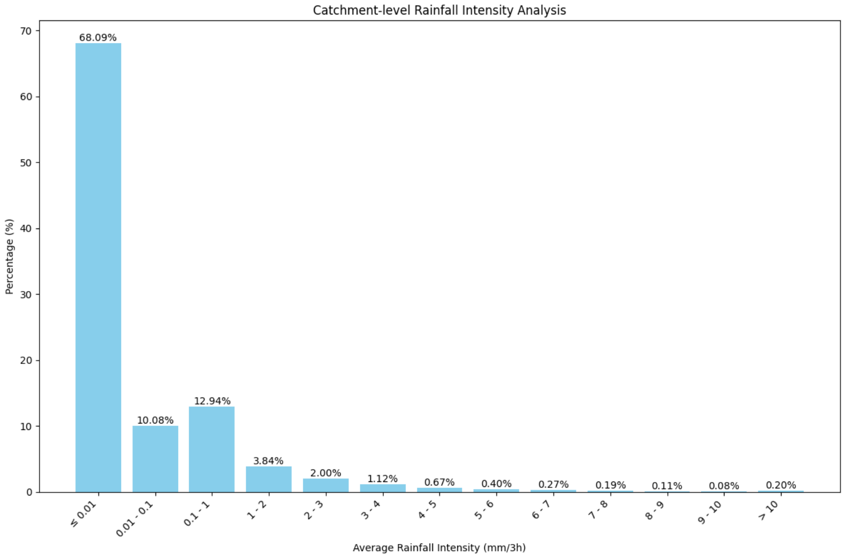


Figure 3.8: Histogram of the precipitation intensities distribution of the selected RT images (Catchment-level).

Conclusion

Therefore, based on the above analysis it can be concluded that the distribution of the rainfall intensities (complete Netherlands region and catchment level) is highly imbalanced (since, majority of the distribution favours low-intensity rainfall). Consequently, this imbalance causes difficulties in fitting the data using standard deep-learning models. To alleviate this problem, additional techniques are necessary (for instance, the incorporation of the EVL loss function). The distributions still have a relatively (compared with exponential distribution) high probability in its tail part, so a heavy-tailed distribution may be required to model the rainfall intensity [27].

3.2 Problem Formulation

The objective of the model is twofold: firstly, to predict skillful precipitation nowcasting results with respect to the whole region of the Netherlands, and secondly, to detect extreme events in the 12 catchment areas, respectively.

In pursuit of the first objective, the aim of this thesis is to perform nowcasting up to three hours ahead, with updates every 30 minutes. This means that each event is represented by a sequence of six images (at T+30, T+60, T+90, T+120, T+150, and T+180 minutes), using the radar images from the preceding 90 minutes (at T-60, T-30, and T minutes) as input. As shown in figure 3.3, each image of an event is of size 256 × 256 since this area covers 90 percent of the land area of the Netherlands and all the relevant catchment locations.

For the second objective, extreme events must be defined within the catchment areas. Given that the scope of the second objective aligns with the problem formulation outlined by Bi et al. [6], this thesis also adopts their definition of extreme events. Usually, extreme rainfall is identified based on the distribution of the maximum annual rainfall. However, with only 14 years of data available, the dataset of yearly maxima is insufficient for effectively training and evaluating models [6]. Consequently, this definition has been relaxed and the precipitation within a catchment is classified as extreme if the average precipitation over a 3-hour period falls within the highest 5% of all measurements recorded from 2008 to 2021. More details on the procedure of selecting events have been mentioned in the following section 3.3.

The reason for choosing extreme events on the basis of rainfall accumulation over 3 hours in the catchment level rather than on the complete Netherlands map is because given an extreme threshold (let's say 10mm/hr) a single pixel value greater than this threshold does not mean there is an extreme-rainfall event. Hazards due to extreme rainfall events are usually caused by heavy and long-term precipitation in certain areas where rainfall gets collected (such as, in the case of catchments). Hence, the event selection process has been carried out on the catchment level with a 3-hour average rainfall accumulation.

3.3 Event Selection

In this study, an event within a catchment is classified as extreme if it results in an average precipitation accumulation exceeding the catchment's specific extreme threshold over a period of 3 hours. Therefore, an extreme event threshold for a certain catchment can be defined using the below-mentioned steps:

1. Identify a catchment as the focal area of study.
2. Categorize each event as either a non-rain event (average catchment precipitation $< 0.1mm/3h$) or a rain event ($\geq 0.1mm/3h$).
3. Organize the events by the average precipitation within the catchment and classify the top 1% as extreme events [6]. This process then sets the extreme threshold for that specific catchment.

However, extreme events (top 1%) alone do not provide an adequate dataset for training a deep learning model. Therefore, events classified within the top 5% of precipitation, referred to as heavy rain events, are also included in the dataset for the catchment. The selected events are split into 3 parts for training, validation, and testing purposes. Events from the years 2008 to 2014 cover the training dataset, 2015 to 2017 cover the validation dataset whereas, 2019 to 2021 cover the testing dataset. Based on this, the total number of events in the training dataset is 30632, whereas the validation dataset contains 3453 events and the testing dataset contains 357 events. The testing dataset only contains all extreme events (i.e., the top 1% events) leading to the least number of events when compared with all the other datasets.

4 Methodology

4.1 Proposed Model

In this section, we describe the model proposed in this thesis in detail. The proposed model - **World Model (EVL)** consists of three distinctive components. The model consists of a Vector-Quantized Variational Autoencoder (VQ-VAE), an Autoregressive Transformer, and a Binary Classifier (also a Transformer). Each component has been described in detail in the subsequent subsections, respectively.

4.1.1 Vector-Quantized Variational Autoencoder

The first component of **World Model (EVL)** is a Vector-Quantized Variational Autoencoder (VQ-VAE). VQ-VAE was first proposed by Oord et al. [28]. VQ-VAE have similar structures when compared to standard autoencoders, i.e., they consist of an encoder and a decoder which learn a forward and reverse mapping from an input space onto a compressed continuous space called the latent space. However, in the case of a VQ-VAE, the continuous latent space is replaced by a discrete latent space [28]. This method is effective in capturing the complex, multi-dimensional features of data. VQ-VAE 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.

Several researchers have used this architecture and improved upon it such as the VQ-VAE model proposed by Esser et al. [29] in their main model framework - VQGAN. The main advantage of the VQ-VAE proposed by Esser et al. is the incorporation of the Perceptual Loss in the main loss function used for training the VQ-VAE. The additional implementation of the Perceptual loss helps the VQ-VAE to learn *perceptually rich* representations of the features present in the original images [29]. The perceptual loss function was proposed in the context of computer vision tasks by Johnson et al. [30]. The authors propose that rather than encouraging the reconstruction of an image $\hat{x}$ to match exactly with the ground truth x, it would be beneficial to have similar feature presentations between the two as computed by loss network ϕ. Therefore, perceptual loss functions are actually deep convolutional neural networks themselves, that have been pre-trained for image classification tasks [30]. One of the most popular choices of ϕ is the pre-trained 16-layer VGG network as suggested by Esser et al. [29], Johnson et al. [30]. Mathematically, the loss function is defined as:

$$\mathcal{L}_{\text{perceptual}}(\hat{x}, x) = \frac{1}{C_j H_j W_j} \left\| \phi_j(\hat{x}) - \phi_j(x) \right\|_2^2 , \tag{4.1}$$

where C_j, H_j, W_j are the dimensions of the feature map $\phi_j(x)$ from the jth layer of the convolutional neural network whereas x and $\hat{x}$ are the target image and output image.

Hence, it can be observed that the perceptual loss function is the Euclidean (squared, normalized) distance between the feature representations of x and $\hat{x}$. Therefore, this is why the VQ-VAE chosen for the model part of this thesis project (**World Model (EVL)**) has been inspired by the research of Esser et al. [29].

The encoder uses downsampling convolutional layers to gradually make the original image smaller in size but richer in detail by increasing the number of channels. After each downsampling layer, there are two Res-Net blocks. These blocks are crucial for enhancing the encoder's ability to recognize and represent the important features of the images, especially as the encoder adds more downsampling layers and becomes deeper (also helps in mitigating the vanishing gradient problem with the increment in the depth of the model). The decoder has a similar structure as the encoder but consists of upsampling layers instead of downsampling layers. A schematic diagram of a standard VQ-VAE has been shown in figure 4.1. In the below-mentioned figure, x is the input radar image of dimensions b = batch, h = height, w = width and t = time (sequence length).

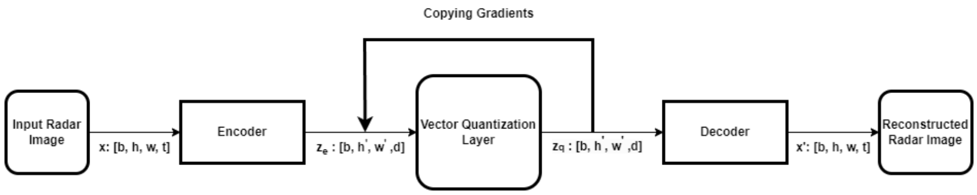


Figure 4.1: Schematic diagram of a VQ-VAE.

The training of the VQ-VAE is done with the help of four loss functions namely, Reconstruction Loss, Codebook Loss, Commitment Loss, and Perceptual Loss. Mathematically, they can be written as:

$$\mathcal{L}(E, D, \mathcal{Z}) = \|x - \hat{x}\|_2^2 + \|\text{sg}[E(x)] - z_{\mathbf{q}}\|_2^2 \quad + \|\text{sg}\,[z_{\mathbf{q}}] - E(x)\|_2^2 + \mathcal{L}_{\text{perceptual}}(\hat{x}, x), \tag{4.2}$$

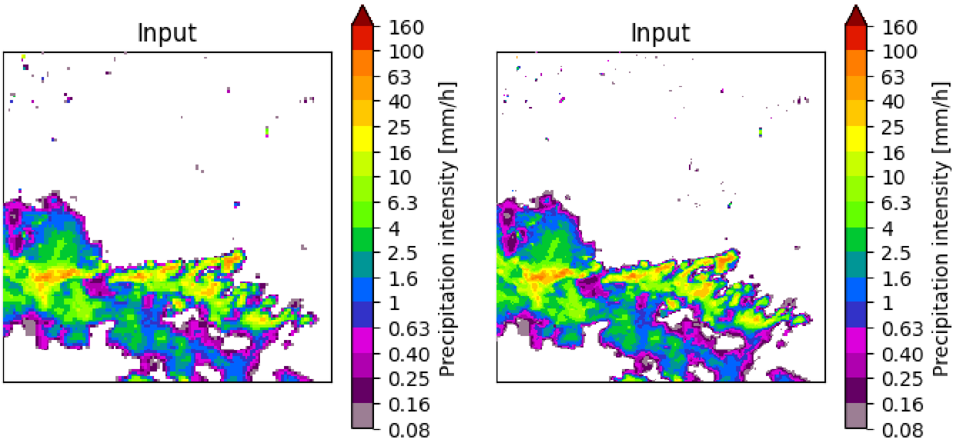
where E, D and $\mathcal{Z}$ represent the Encoder, Decoder and the Codebook. $z_{\mathbf{e}} = E(x) \in \mathbb{R}^{h \times w \times n_z}$ represents the encoded image while $\hat{x} = D(z_q)$ is the reconstructed image using $z_{\mathbf{q}}$. We obtain $z_{\mathbf{q}}$ using an element-wise quantization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

$$z_{\mathbf{q}} = \mathbf{q}(z_{\mathbf{e}}) := \left(\underset{z_k \in \mathcal{Z}}{\arg\min} \|\hat{z}_{ij} - z_k\| \right) \in \mathbb{R}^{h \times w \times n_z}. \tag{4.3}$$

In the above equation (4.2), the term $\|x - \hat{x}\|_2^2$ is the reconstruction loss that optimizes the encoder and the decoder. The terms $\|\text{sg}[E(x)] - z_{\mathbf{q}}\|_2^2 + \|\text{sg}\,[z_{\mathbf{q}}] - E(x)\|_2^2$ together comprise the codebook loss and the commitment loss which ensure that the encoder reliably produces outputs close to the codebook vectors. The term $\mathcal{L}_{\text{perceptual}}(\hat{x}, x)$ is the perceptual loss which helps the VQ-VAE learn perceptually rich features (shown in equation (4.1)).

The Vector Quantization layer as shown in figure 4.1 is responsible for the quantization of the latent space generated by the Encoder of the VQ-VAE. It takes $z_{\mathbf{e}}$ and selects the closest embedding from the Codebook (based on Euclidean distance) and outputs $z_{\mathbf{q}}$. Acknowledging that backpropagation is infeasible through the *arg min* operation in equation (4.3), gradients are instead propagated by approximating the gradients using the stop gradient operator sg, from $z_{\mathbf{q}}$ to $z_{\mathbf{e}}$. This approach does not directly minimize the loss function but allows the transfer of some gradient information back for model training.

Due to the constraints on computational efficiency, the input radar images have first been downsampled to a spatial resolution of 128 × 128 from 256 × 256. The downsampling does not change the semantic information represented by the input radar maps as shown in the below-mentioned figure 4.2. Moreover, it helps in efficient memory usage of the Graphical Processing Unit (GPU) while training the VQ-VAE in the achievement of a lower dimensional latent space (8 × 8 × 1024).



(a) (b)

Figure 4.2: Representation of an input radar image in the training dataset with spatial resolution (a) 128 × 128 and (b) 256 × 256.

As mentioned-above already, the encoder consists of 5 downsampling layers (2D Convolutional layers) each containing 2 ResNet blocks in between them. This reduces the spatial dimension of the input radar images to the following resolutions: 128 → 64 → 32 → 16 → 8. Furthermore, the last stage of the encoder includes an attention block used to capture the relationships between features before the quantization step. The decoder mirrors the structure of the encoder but consists of upsampling layers instead of downsampling layers to reconstruct the image from the discrete codes to its original spatial resolution as 8 → 16 → 32 → 64 → 128.

Therefore, with the help of the VQ-VAE, each radar image is converted into a sequence of 64 discrete codes. The codebook used for this conversion contains 1024 unique tokens, meaning each token can be an integer ranging from 0 to 1023 (each token is an index in the codebook). Additionally, the embedding dimension, denoted as n_z in equation (4.3), has the value 1024 as mentioned above. This setup allows for a detailed representation of each image within a high-dimensional embedding space. Examples of the reconstructed precipitation fields with respect to the ground truth images have been displayed in figures A.1 and A.2 in Appendix A.

4.1.2 Autoregressive Transformer

The transformer model uniquely utilizes a mechanism called attention to understand the relationship between different parts of its input without considering the relative position of each part with respect to the others [31]. Recent research by Esser et al. [29], Yan et al. [32] has displayed that, although transformers were initially created for tasks involving Natural Language Processing (NLP), they are indeed substantially effective in computer vision tasks that involve the processing of image and video data. Bi et al. [6] also implement an autoregressive transformer in their research for extreme precipitation nowcasting purposes.

The architecture of a transformer mainly consists of two components: an encoder and a decoder. These components are made up of attention layers as well as fully-connected layers that incorporate the attention mechanism amongst the different parts of the input. Additionally, this setup of stacked layers is complemented by position-wise feed-forward networks, which process each part of the input sequence independently. Moreover, the feed-forward network is also responsible for introducing non-linearity in the processing of the input sequence (GELU has been utilized for the proposed model - **World Model (EVL)**). This helps in learning more complex patterns which are typically not achievable using linear transformations [31]. The below-mentioned figure 4.3 shows a structure of the transformer implemented in **World Model (EVL)**. As evident from the figure, positional encoding is also added to the input provided to the auto-regressive transformer. Positional Encoding is necessary since the attention mechanism of the transformer treats each input token (the embedded input sequence) equally, irrespective of their positions or the sequence in which they appear. Hence, to add crucial information about the sequence order to the model's input, positional encoding is incorporated.

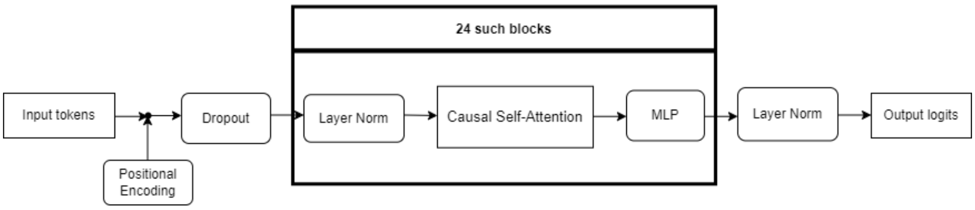


Figure 4.3: Structure of the Autoregressive Transformer.

The self-attention mechanism within a transformer processes these input tokens (embedded with positional encoding) through three position-wise linear layers, generating three sets of representations namely: Queries (Q), Keys (K), and Values (V). The attention weights are calculated using equation (4.4) where Q, K, and V can be calculated as $Q = XW_Q$, $K = XW_K$, and $V = XW_V$ and d_k is the dimensionality of Q, K and V. The model learns the weight matrices $W_Q, W_K$and, W_V through the backpropagation on the transformer's loss function. The incorporation of $\sqrt{d_k}$ as a scaling value is done since the dot product of high-dimensional matrices would produce large values which could prove to be problematic when applying the softmax function (as it is sensitive to large values).

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V, \tag{4.4}$$

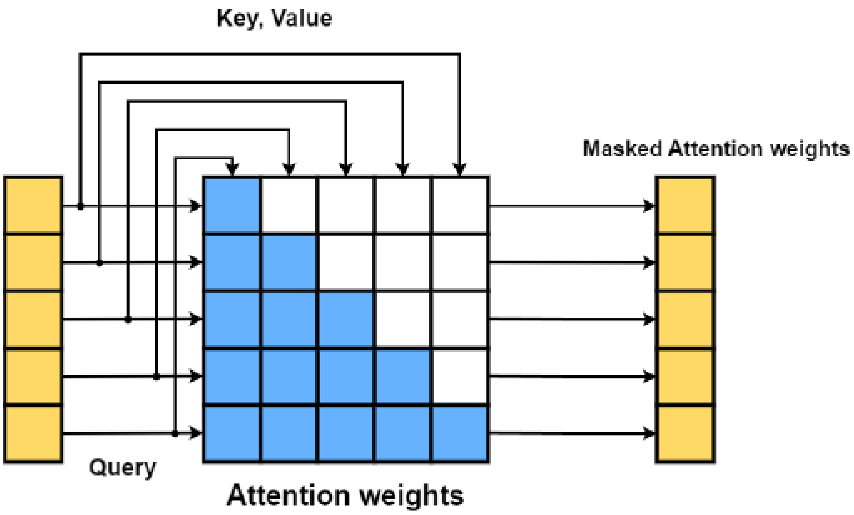


Figure 4.4: Representation of Causal Attention mechanism.

The utilization of the *Softmax* function is to obtain a set of weights that accurately evaluate each token in relation to the global context. The Softmax function helps in the normalization of the attention scores ($\frac{QK^T}{\sqrt{d_k}}$) and ensures that the scores follow a probability distribution with the weights of each token summing up to 1 as shown in equation (4.5).

$$\text{softmax}\left(\frac{Q_iK^T}{\sqrt{d_k}}\right) = \frac{e^{\left(\frac{Q_iK^T}{\sqrt{d_k}}\right)}}{\sum_{j=1}^{d_k} e^{\left(\frac{Q_jK^T}{\sqrt{d_k}}\right)}}, \tag{4.5}$$

Moreover, the model uses a causal self-attention mechanism, as shown in Figure 4.4. This approach involves using masking to ensure that the generation process is autoregressive, meaning the model makes predictions based on previously seen and current tokens only. This is achieved by blocking out (masking) the upper part of the attention weights matrix (shown as white blocks in the figure 4.4) to prevent the transformer from being influenced by tokens that appear later in the sequence. Then, the transformer applies a straightforward, element-by-element transformation to obtain logits, which help in predicting the next element in the sequence, thereby preserving the natural sequence order of the tokens.

As shown in figure 4.3, the autoregressive transformer utilizes 24 layers of self-attention (as the model has 24 such blocks). The input to the transformer is the discrete latent representation generated by the VQ-VAE for the original radar images (also referred to as, *tokens*). The transformer effectively learns the distribution of the tokens by utilizing a cross-entropy loss function. The training stage and the generation stage of the autoregressive transformer have been described in the subsequent sections.

Training Stage

During the training stage, the original radar input images are transformed into quantized codes through the vector quantization layer (as shown in figure 4.1). This representation is given by $\mathbf{z}_q = q(E(x))$ where E is the Encoder of the VQ-VAE. This step produces a sequence of codes, within the range of 0 to $|\mathcal{Z}| - 1$, representing the indices from the VQ-VAE codebook, for every encoded input image. These indices or *tokens* are then passed through an embedder that transforms the corresponding discrete tokens into continuous vectors. As shown in figure 4.3, these *Input sequence of tokens* are then added with positional embeddings which inject the sequence order information into the vector representations of the respective tokens. The transformer processes these embedded-positional encoded tokens through all the layers and outputs raw, unnormalized predictions for each token known as logits. These logits are evaluated against the actual distribution of the tokens using a cross-entropy loss function. The cross-entropy loss function when implemented in the PyTorch framework, innately converts the logits into probabilities using a softmax layer, hence, accepting probabilities as its inputs. The cross-entropy loss function can be written as:

$$\mathcal{L}_{\text{Transformer}} = \mathbb{E}_{x \sim p(x)}[-\log p(\mathbf{z})], \tag{4.6}$$

where $p(\mathbf{z}) = \prod_{i=1}^{N} p\left(z_i \mid z_{<i}\right)$, indicates that, given the indices (tokens) $z_{<i}$, the transformer is trained to predict the distribution of the next possible indices z_i.

Generation Stage

In the generation stage, the transformer is given the conditional input tokens that serve as the starting point for generating new tokens. A Key-Value (KV) cache is used here to speed up the process at every step. KV cache is done by keeping the keys and values that were determined during the self-attention phase, allowing the model to avoid redoing calculations for these elements in future steps. This is especially beneficial in autoregressive models, where the transformer predicts one token at a time. The KV cache enables each newly produced token to depend on tokens that have already been generated without having to recalculate the entire attention map, thereby greatly improving the efficiency of the generation process. The decoder of the transformer carries out KV caching since it is responsible for the generation of the new tokens (based on the conditional ones) [33]. KV caching requires more computational capacity to store the Keys and Values of the previous tokens. However, it improves the overall efficiency of the generation phase as mentioned above.

Furthermore, in the generation stage, the logits of the last token are sliced and concatenated with the previous tokens to form a new sequence. The autoregressive transformer is designed to continue this production of logits for a total number of steps determined by the number of prediction frames multiplied by the total number of tokens required to encode the input radar image. Once, the logits for the entire sequence of prediction frames have been generated, top-k and top-p sampling techniques are applied to narrow down the sampling pool to the k most likely tokens or to a subset of tokens that together add up to a specified probability p, enhancing the quality and coherence of the generated tokens for the entire sequence.

For the final output, categorical sampling has been utilized that allows the selection of certain a token from the predicted probability distribution by the model. Once, the entire sequence of tokens has been predicted by the transformer. The tokens are fed back to the decoder of the VQ-VAE, which decodes the discrete sequence of tokens into continuous representations from the Codebook of the VQ-VAE, forming a predicted radar map.

4.2 Addressing Extreme Events

As already mentioned in section 4.1.1, the VQ-VAE transforms the original radar input images into discrete latent space representations by quantization. The discrete representations or tokens represent different precipitation areas on the input radar maps. Consequently, these tokens correspond to different levels of rainfall namely: no rain, light rain, extreme rain on the original radar input map. However, as mentioned in Chapter 3, instances of low and moderate rainfall outnumber the occurrence of heavy or extreme rainfall. Therefore, it can be said that the number of extreme tokens would be less when compared to the number of tokens that capture light or no rainfall. This observation indicates that the autoregressive transformer is being trained on a significantly imbalanced dataset. Due to the lack of adequate samples of extreme precipitation patterns in the training dataset, the autoregressive transformer's output distribution is likely to skew towards more frequently occurring tokens as well,i.e., those signifying no rainfall or light rainfall. To address this problem, a binary classifier has been incorporated that classifies the tokens into *extreme* or *non-extreme*, and then an additional loss function known as EVL (equation (2.30)), has been incorporated inside the autoregressive transformer as a regularizer. The Binary classifier and the incorporation of the EVL loss function have been explained in detail in the subsequent sections.

4.2.1 Binary Classifier

For the classification of the tokens into extreme or non-extreme, a transformer is incorporated along with the autoregressive transformer. The input to this transformer is the sequence of tokens (for the corresponding encoded radar maps) that are generated from the auto-regressive transformer during its training phase. The classifier is trained using a standard binary cross entropy loss function given by:

$$\mathrm{BCE}\left(u_t, v_t\right) = -\, v_t \log\left(u_t\right) - \left(1 - v_t\right) \log\left(1 - u_t\right), \tag{4.7}$$

where u_t is the predicted probability and v_t is the ground truth label. The ground truth labels v_t are calculated based on the averaged precipitation over a threshold of 5mm on the input radar maps. This allows the classification of all the tokens corresponding to an extreme/non-extreme event based on the ground truth labels. The classifier generates logits for the two aforementioned classes which are then passed through a SoftMax layer to generate the predicted probabilities u_t.

Therefore, the predicted probabilities u_t and the ground truth labels v_t act as the input to the EVL loss function that has been derived in section 2.2.3. The EVL loss function is given as:

$$\begin{aligned} \mathrm{EVL}\left(u_t, v_t\right) = &-\beta_1 \left[1 - \frac{u_t}{\gamma}\right]^{\gamma} v_t \log\left(u_t\right) \\ &- \beta_0 \left[1 - \frac{1 - u_t}{\gamma}\right]^{\gamma} \left(1 - v_t\right) \log\left(1 - u_t\right). \end{aligned} \tag{4.8}$$

where $\Pr(v_t = 1) = \beta_1$ and $\Pr(v_t = 0) = \beta_0$ are the proportions of the extreme events and normal events. γ is the shape parameter as mentioned in section 2.2.3. However, several researchers also term it as the *Extreme Value Index* [6], [7]. The EVL loss function is incorporated with the loss function of the autoregressive transformer (equation (4.6)) as an additional regularizer. Therefore, the updated loss function on which the autoregressive transformer is being trained on, can be written as:

$$\mathcal{L}_{\text{Transformer-EVL}} = \mathcal{L}_{\text{Transformer}} + \lambda[\text{EVL}(u_t, v_t)]. \tag{4.9}$$

The values for β_0 and β_1 are taken as 0.95 and 0.05 respectively since top 5% of the events are considered as extreme events. The value of γ for EVL was set to 1, as this setting demonstrated optimal performance by Bi et al. [6]. The value of λ has been chosen as 1 based upon the analysis of results shown in Appendix C. Therefore, the model **World Model (EVL)** consists of these three components i.e. the VQ-VAE, the autoregressive transformer, and the Binary classifier whereas the baseline **World Model** consists of only the VQ-VAE and the Autoregressive Transformer (without the implementation of the EVL loss function).

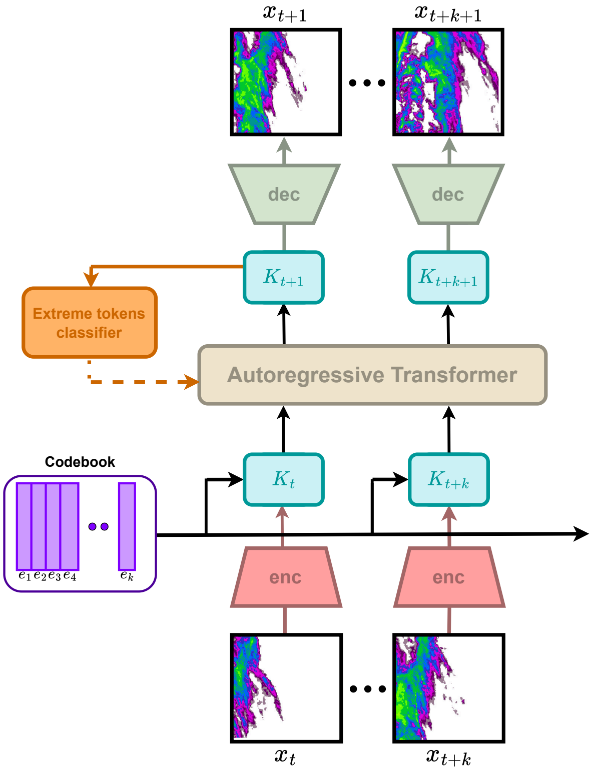


Figure 4.5: Representation of the entire model - **World Mode (EVL)**.

In figure 4.5, the representation of the **World Model (EVL)** has been shown which consists of the three main components described above sections. The components of the VQ-VAE are shown as the *enc*, *dec*, and the *codebook* blocks whereas the Autoregressive Transformer is self-explanatory. The input tokens for the encoded radar input images are passed to the *Extreme tokens classifier* block which classifies the tokens as *extreme/non-extreme* and gets simultaneously trained with the autoregressive transformer.

4.2.2 Post-processing Technique

Since the input radar maps (RT-dataset) do not comprise of high-intensity precipitation pixels, Chen et al. [34] proposed a post-processing technique for a better emphasis on high-intensity precipitation pixels during nowcasting. The method is expressed using the equation below:

$$TP[i][j] = \left(1 + a\left(\frac{RP[i][j]}{\max(RP)}\right)^{b}\right) * RP[i][j], \tag{4.10}$$

where TP and RP are the post-processed and unprocessed predictions, respectively. Moreover, (i, j) represents the location of the pixels in the prediction maps. The parameters a and b are determined to reach the maximum Gilbert Skills Score (GSS) on the validation dataset (the 357 extreme events). Bi et al. [6] use the values $a = 0.66$ and $b = 0.81$ and the same has been applied in the scope of this thesis work to be consistent with the application of it.

The aforementioned post-processing technique enhances the detection rate of high-intensity precipitation pixels. However, in doing so it also increases the number of False Alarm cases so it is applied in tandem with the ensemble technique (as shown in section 5.1.2).

4.3 Experiment Configuration details

In this section, the details about the training configuration of the model **World Model (EVL)** are described in detail. As mentioned above, the model consists of three main components: VQ-VAE, an Autoregressive Transformer, and a Binary classifier for classifying the extreme and non-extreme tokens. The following list contains the details of the hyper-parameters controlling the architecture of the different components of the model as well as their training procedures.

1. VQVAE model configuration:
   - Total Number of down-sampling layers (2D Convolutional Layers): 5
   - Size of codebook: 1024
   - Embedding dimension of each token in the codebook: 1024
   - Number of ResNet Blocks after every down-sampling layer: 2
2. VQVAE training configuration:
   - Learning rate: 0.0001
   - Batch size: 64
   - Weight decay: 0.1
3. Auto-regressive Transformer model configuration:
   - Embedding dimension: 1024
   - Number of attention layers: 24
   - Number of attention heads: 16
4. Auto-regressive Transformer training configuration:
   - Learning rate: 0.0001

- Batch size: 64
- Weight decay: 0.01
- Drop out rate (for both attention and fully connected layer): 0.1

5. Binary Classifier Transformer model configuration:
    - Embedding dimension: 1024
    - Number of attention layers: 6
    - Number of attention heads: 8
6. Binary Classifier training configuration:
    - Learning rate: 0.0001
    - Batch size: 64
    - Weight decay: 0.01

The main codebase for the model has been developed in the PyTorch framework. The training as well as the different experiments (detailed in Chapter 5) are conducted on an NVIDIA RTX A6000 GPU. For the training of all the components of the models, 16-bit Automatic Mixed Precision (AMP) using a library called *Fabric* in the PyTorch framework [35] has been implemented. Traditionally, the training of Deep neural networks (DNNs) has relied upon IEEE single-precision format [36], however, this leads to an increment in training time especially large architectures like the different components of the **World Model(EVL)** model. However, AMP enables the model (weights of the different layers) to be trained on lesser precision while maintaining the accuracy of the DNNs achieved with single precision. AMP helps speed up the training process of the DNNs by reducing memory requirements [36]. This helps in the overall improvement of the training time of the model when compared with the other models as shown in Appendix B.

Moreover, to reduce the training time even further, the input radar images have been converted into NumPy arrays and saved locally. This is mainly done to reduce the time in calling the input images from the Radar folder path (as they are saved in .h5py format, available on the KNMI website [25]). Since the training dataset consists of 30632 sequences of radar images where each sequence comprises of 9 images, it can increase the training time if every time the image path has to be called for training. This conversion of the input radar images into NumPy array mainly helps in improving the training efficiency of the VQ-VAE as it is trained on the images itself. The training time significantly improved from approximately 30hrs/epoch to 3.5hrs/epoch in the case of the VQ-VAE. As for the Autoregressive Transformer, since it is trained on tokens-level representations of the encoded images, the training time is mainly improved because of the application of the 16-bit AMP technique mentioned above.

4.4 Verification Metrics

In this section, the details about the different verification metrics used to evaluate the predictions generated by the various models have been elaborated. The first two sub-sections contain information on the metrics used to evaluate the nowcasting performance for the entire region of the Netherlands. In contrast, the last sub-section contains details regarding the metrics used for evaluating the extreme event detection performance in the catchment areas.

4.4.1 Continuous Verification metrics

Pearson's Correlation Coefficient (PCC)

Pearson's Correlation Coefficient (PCC) is utilized as an index to ascertain the correlation between two disparate datasets. The coefficient, symbolized by ρ, is derived according to the following equation:

$$\rho = \frac{1}{N_f} \sum_{i=1}^{N_f} \left(\frac{F_i - \mu_F}{\sigma_F} \right) \left(\frac{O_i - \mu_O}{\sigma_O} \right), \tag{4.11}$$

In this equation, F_i and O_i represent the amounts of rainfall in a specific cell of the forecast and observation maps, respectively. The variables μ_F and μ_O denote the mean rainfall values from the forecast and observation frames, respectively, while σ_F and σ_O are their standard deviations. The term N_f signifies the total count of pixels within the radar map for a specific forecast lead time.

A higher PCC is an indication of a stronger correlation between the two images, with a PCC of 1 being the ideal, denoting perfect correlation. However, this is often unattainable in practical scenarios. Some research [2] has indicated that a PCC value below the threshold of the reciprocal of e (approximately 0.37) is indicative of a forecast lacking skill. Within the scope of this thesis, PCC will be evaluated for lead times of 30, 60, 90, 120, 150, and 180 minutes.

Mean Absolute Error (MAE)

For the evaluation of the predictions, Mean Absolute Error (MAE) has also been utilized as one of the continuous metrics. MAE measures the absolute value *(l1-norm)* of the error between the ground truth and the prediction frames and is given by:

$$MAE = \frac{\sum_{i=1}^{N_f} |F_i - O_i|}{N_f}, \tag{4.12}$$

where F_i and O_i represent the amount of rainfall in a certain cell in the prediction and the ground truth maps, and N_f, represents the total number of pixels in the respective map. Similar to PCC, the MAE metric is also calculated for all the lead times i.e. 30, 60, 90, 120, 150, and 180 minutes. A smaller Mean Absolute Error (MAE) between the forecast and observation indicates a higher accuracy of the predicted frame.

4.4.2 Spatial Verification metric

Fractional Skills Score

The Fractional Skill Score (FSS) is a spatial metric utilized for evaluating the precision of precipitation forecasts. By varying the length scale n, which impacts the extent of the area considered, different FSS scores can be calculated giving us the analysis of predictions with a more extensive area used in the calculation of the score. Generally, a larger n leads to better FSS scores. The FSS value is confined between 0 and 1, where a higher FSS indicates a more accurate forecast map. The FSS is defined as:

$$FSS = 1 - \frac{MSE(n)}{MSE_{\text{ref}}(n)}, \tag{4.13}$$

In this context, $MSE(n)$ represents the mean square error across observations and forecasts at a specified length scale n, whereas $MSE_{\text{ref}}(n)$ denotes the maximum MSE across both observations and forecasts at the same length scale. The $MSE_{\text{ref}}(n)$ is computed as follows:

$$MSE_{\text{ref}}(n) = \frac{1}{N_x N_y} \left(\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} O_{i,j}^2(n) + \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} F_{i,j}^2(n) \right), \tag{4.14}$$

Here, N_x and N_y denote the total number of columns and rows present in the ground truth and forecast maps, respectively. $F_{i,j}^2(n)$ and $O_{i,j}^2(n)$ are the squared fractions of the forecast and observation greater than the defined rainfall threshold for grid cell (i, j), considering adjacent points within a neighborhood up to n. These fractions are calculated as:

$$O_{i,j}^2(n) = \frac{1}{n^2} \sum_{k=1}^{n} \sum_{l=1}^{n} I_o \left(i + k - 1 - \frac{n-1}{2}, j + l - 1 - \frac{n-1}{2} \right), \tag{4.15}$$

where I_o is a binary field indicating whether the rainfall at a certain map location surpasses a given threshold. The sum of this field denotes the number of cells surpassing the rainfall criterion within a specific grid cell, and this sum is normalized by the total number of cells within the $n \times n$ area. Predictions are typically considered skillful forecasts when the FSS exceeds $0.5 + \frac{f_o}{2}$, where f_o is the domain-averaged proportion of observed rainfall [8].

4.4.3 Categorical Verification Metrics

In the assessment of categorical metrics, each pixel in the prediction and observation maps is initially categorized as either positive (greater or equal to) or negative (below) a specific threshold [37]. Subsequently, pixels are allocated into one of four distinct categories:

1. H: true positive, where both observation and prediction are positive.
2. M: false negative, where the observation is positive, but the prediction is negative.
3. F: false positive, where the observation is negative, but the prediction is positive.
4. R: true negative, where both observation and prediction are negative.

The Critical Success Index (CSI), prevalent within the nowcasting community, encapsulates the binary classification performance by enhancing precision and simultaneously penalizing false alarms. An elevated CSI correlates to enhanced performance and is computed as:

Event predicted	Event observed		Total
	Positive	Negative	
Positive	Hits (H)	False alarms (F)	Predicted positives
Negative	Misses (M)	Correct negatives (R)	Predicted negatives
Total	Observed positives	Observed negatives	Total

Table 4.1: Confusion Matrix for categorical metrics.

$$CSI = \frac{H}{H+F+M}. \tag{4.16}$$

The False Alarm Ratio (FAR) is another pivotal metric for binary classification efficacy, regularly utilized within weather forecasting. FAR evaluates the precision of predictive alarms and is inversely related to performance i.e. a lower FAR indicates a superior performance in classification task. FAR is calculated by:

$$FAR = \frac{F}{F+H}. \tag{4.17}$$

These categorical metrics have been used to evaluate the accuracy of the predictions generated by the different models, with respect to the ground truth maps in Chapter 5.

4.4.4 Catchment Verification Metric

To check how well a model can identify extreme weather in a catchment area, we use certain methods. These extreme weather events are classified as either happening or not, similar to other types of classification. Events are categorized as true positive (H), false negative (M), false positive (F), and true negative (R), depending on whether they exceed certain thresholds. The Critical Success Index (CSI) and the False Alarm Ratio (FAR) are used to check if the classification is correct. Another method is the Receiver Operating Characteristic (ROC) curve, which is created by using different thresholds for extreme events. Since there is an imbalance in the observations between the two classes (the number of extreme events is significantly lesser than the non-extreme events) another analysis is performed using the Precision-Recall curve [38]. The Precision-Recall curve is also created using different thresholds for extreme events. The choice of the thresholds for both ROC and Precision-Recall curves has been explained in detail in section 5.2.2.

Receiver Operating Characteristic (ROC) Curve

The ROC curve is constructed by calculating two things: the Hit Rate (HR) and the False Alarm rate (FA). They are given by:

$$HR = \frac{H}{H+M}. \tag{4.18}$$

$$FA = \frac{F}{F+R}. \tag{4.19}$$

By choosing different thresholds, we can get different values of HR and FA. We plot these values on a graph with FA on the x-axis and HR on the y-axis. Connecting these points gives us the ROC curve. This curve is a useful tool to compare how well different models can detect and classify events. The area under the ROC curve (AUC) is often used to measure detection ability. A bigger AUC means the model is better at detecting events.

Precision-Recall Curve

The Precision-Recall curve is constructed using two metrics: Precision and Recall (also known as the Hit Rate, as mentioned above). The formulae for both of these metrics are given as:

$$Precision = \frac{H}{H+F}. \tag{4.20}$$

$$Recall = \frac{H}{H+M}. \tag{4.21}$$

The Precision-Recall Curve is also constructed in a similar way as the ROC curve i.e. by choosing different thresholds, we calculate multiple precision and recall scores. The precision scores are plotted on the y-axis whereas the recall scores are plotted on the x-axis. A higher AUC for a certain model signifies a better performance of the model in the detection of positive events with a lesser number of false negatives.

5 Experiments and Results

Based on the objectives of this thesis project, the evaluation of the model can be divided into two categories i.e. Nowcasting performance of the different models on the whole region of the Netherlands and Extreme Event detection performance on the catchment regions. As mentioned in Chapter 3, Catchments are those regions where precipitation accumulates over a period of time making them susceptible to floods (during heavy rainfall).

In the initial section of this chapter, the outputs of the model and its real-time forecasting capabilities for the entire area under investigation are assessed. The evaluation is done by employing a range of continuous as well as categorical metrics such as PCC, MAE, FSS, CSI, FAR which, are defined in the last section of the previous chapter.

The second section focuses on identifying severe weather events. Here, the average precipitation accumulation over a 3-hour period in a specific catchment area is calculated using data from the preceding forecasting outcomes. This calculation is then compared with the established threshold for extreme events. Subsequently, each incident is categorized into one of four scenarios (true positive, false positive, true negative, false negative) based on this comparison. The study then utilizes various standard categorical metrics for binary classification (HR, FA, CSI, FAR, AUC of ROC curve, and Precision-Recall curve) to assess the model's performance in detecting extreme weather events. For both segments of the study, the performance of PySTEPs is used as a comparative standard. The comprehensive setup of PySTEPs is described in section 2.1.1. Moreover, PySTEPS is also provided with the same input as that used in the deep learning models which are a part of this study.

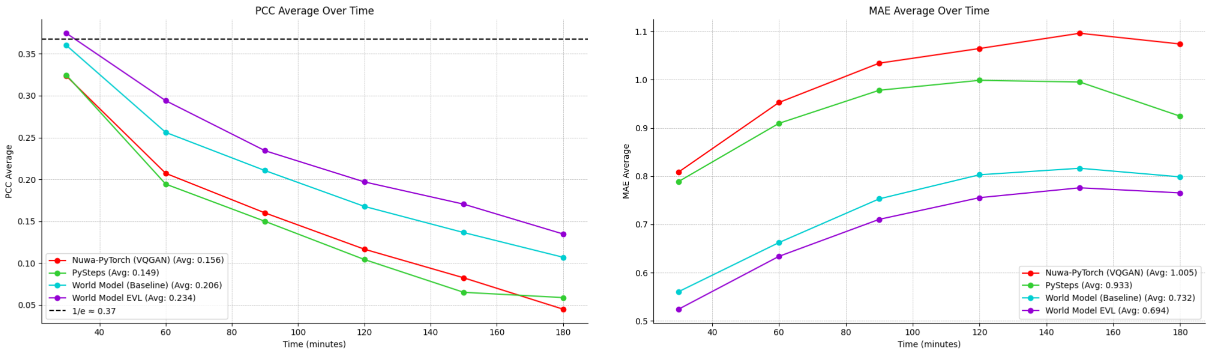
5.1 Nowcasting performance in the whole region of Netherlands

5.1.1 Analysis of models without ensemble and post-processing technique

In this section, nowcasting performance of four models is compared as well as analyzed. The first model is the benchmark **Nuwa-Pytorch (VQGAN)** model proposed by Bi et. al [6], which consists of a Vector-Quantized Generative Adversarial Network (VQGAN) along with an auto-regressive transformer which has the EVL loss function incorporated in it. The second model is **World Model (EVL)** which is the main model of this thesis project consisting of a VQVAE (Vector-Quantized Variational Autoencoder) along with an auto-regressive Transformer (where the EVL loss function is also incorporated) and a binary classifier (similar to a Vision Transformer) for consistent classification of extreme and non-extreme tokens. The third model used in this comparative study, consists only of a VQVAE and an auto-regressive Transformer, termed as **World Model** (also acts as a baseline for **World Model (EVL)**). Lastly, PySTEPS also has been utilized in this study, to understand the performance of state-of-the-art models with respect to Deep generative models such as the ones mentioned above.

In the below-mentioned figure 5.1, two continuous metrics are analyzed for the different models with respect to different lead times. The first sub-figure 5.1a, shows the PCC values for different models over different lead times whereas the second sub-figure 5.1b shows the MAE metrics.

Generally, a prediction is deemed effective if the PCC is greater than $\frac{1}{e}$. This helps in figuring out the longest time a model can predict accurately before the correlation fades, also known as the maximum skillful lead time. The PCC plot indicates that most models can predict accurately for less than 30 minutes, mainly because the area studied is quite small. **World Model-EVL** shows the highest PCC average metric when compared with other models and, also has a *skillful prediction* for the first lead time (30 minutes). The baseline - **World Model** also shows better performance than PySTEPS and **Nuwa-PyTorch (VQGAN)**. The PCC values show a decreasing trend over the whole horizon (all the lead times) since the predictions become less accurate with an increment in lead time (leading to a lesser correlation with respect to the ground-truth frames). Moreover, the **World Model-EVL** model also shows a significant improvement in PCC across all the lead times when compared with the **Nuwa-PyTorch (VQGAN)** model (an overall 45 percent increment in the average PCC value).



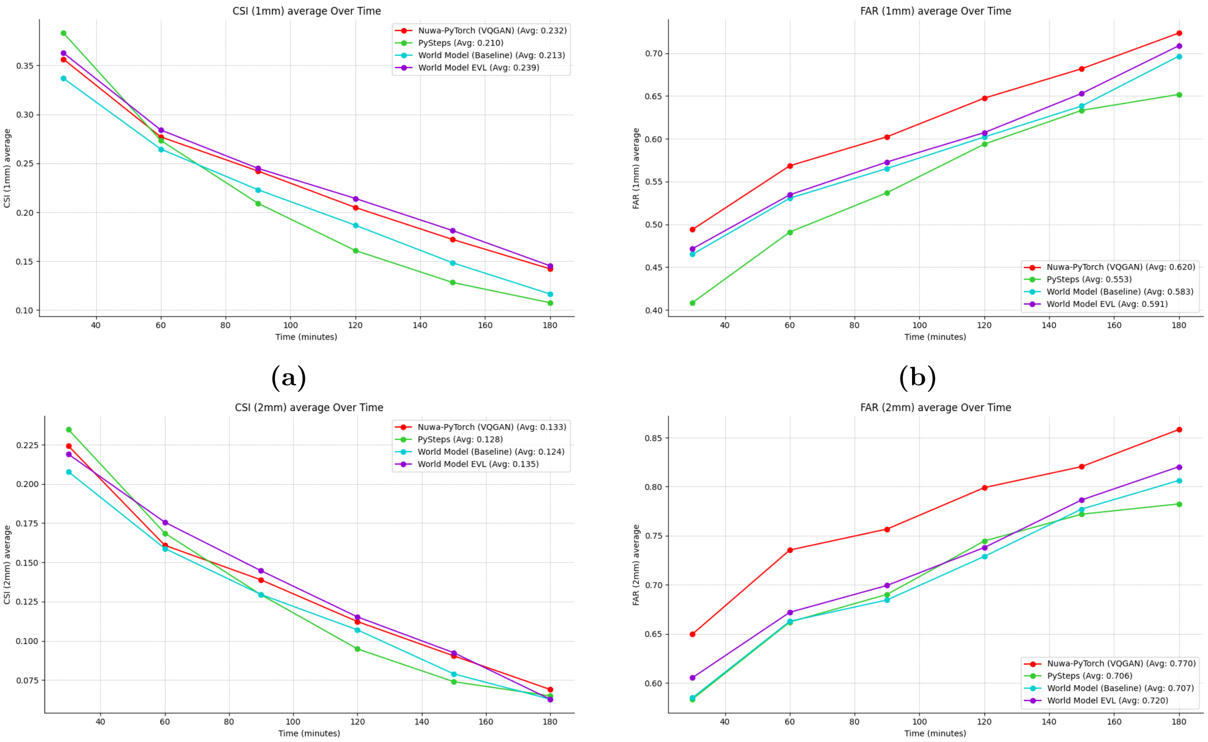
(a)

(b)

Figure 5.1: 3-hour nowcasting performance verification: continuous metrics (sub-figure (a) for PCC and sub-figure (b) for MAE). Relationship between lead time (mins) and metric scores, with 3-hour averaged scores shown in the legend.

Regarding MAE, a high MAE often occurs when the model is not able to predict the overall map correctly, when compared with the ground truth frames or when the prediction frames do not capture the correct precipitation intensity of a certain pixel in a respective frame. A lower MAE value typically indicates a better prediction performance since it signifies a smaller deviation from the ground-truth frames. All the models show a similar upward trend of MAE across the whole horizon. This is because of the increment in uncertainty associated with predictions further out into the future. The **World Model-EVL** outperforms all others with the lowest starting MAE of approximately 0.52, and while it follows the common trend of an increasing MAE over time, it maintains the lowest error across all time points. It peaks at around 120 minutes with an MAE of around 0.75 and then slightly decreases, remaining relatively flat until 180 minutes. The average MAE of 0.694 indicates that this model is the most accurate among the other models in this experiment. Moreover, the **World Model** model (baseline) also shows a significant improvement in the average MAE, when compared with PySTEPS (approximately, a 22%

decrement) and **Nuwa-PyTorch (VQGAN)** (approximately, a 27% decrement).



(a) **(b)**

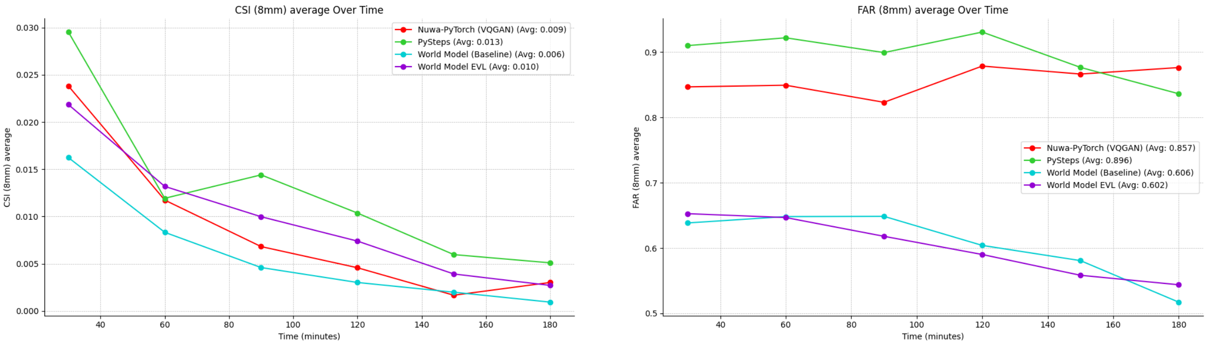
(c) **(d)**

Figure 5.2: 3-hour Nowcasting performance verification on the whole Netherlands region: Categorical Scores - CSI and FAR for a, b) 1mm and c, d) 2mm. Relation between lead time and metric scores, with 3-hour averaged scores shown in the legend.

In the above-mentioned figure 5.2, the Critical Success Index (CSI) and the False Alarm Ratio (FAR) have been plotted for the four models with the thresholds of 1mm, 2mm to analyze the performance of the models. The application of the different thresholds is to assess the performance of the respective models in predicting accurate precipitation intensities with regard to light rainfall (1mm & 2mm). In the field of meteorology, the Critical Success Index (CSI) is a widely recognized metric that summarizes the accuracy of predicting whether certain events, such as rainfall, will surpass a specific threshold or not. Besides CSI, FAR has also been used to assess the detection ability of the corresponding models.

From sub-figure 5.2a and 5.2c, it can be observed that **World model-EVL)** has the highest CSI (1mm) and CSI (2mm) average compared to all the other models. However, it can also be observed that **Nuwa-PyTorch (VQGAN)** shows almost similar performance with respect to **World Model-EVL**. This highlights the significance of the incorporation of the EVL loss function in the auto-regressive transformers of the respective models. Moreover, the baseline **World Model** also displays similar performance when compared with PySTEPS.

From sub-figure 5.2b and 5.2d, it can be observed that in both the figures, **Nuwa-PyTorch (VQGAN)** has the highest average FAR metric when compared with the other models suggesting, that the model predicts more false alarms with respect to others. Also, PySTEPS has the lowest FAR average for both the thresholds (1mm & 2mm). However, the average FAR metric for PySTEPS is comparable to the average values of both **World Model** and **World Model-EVL** - (especially for the 2mm threshold CSI and FAR).



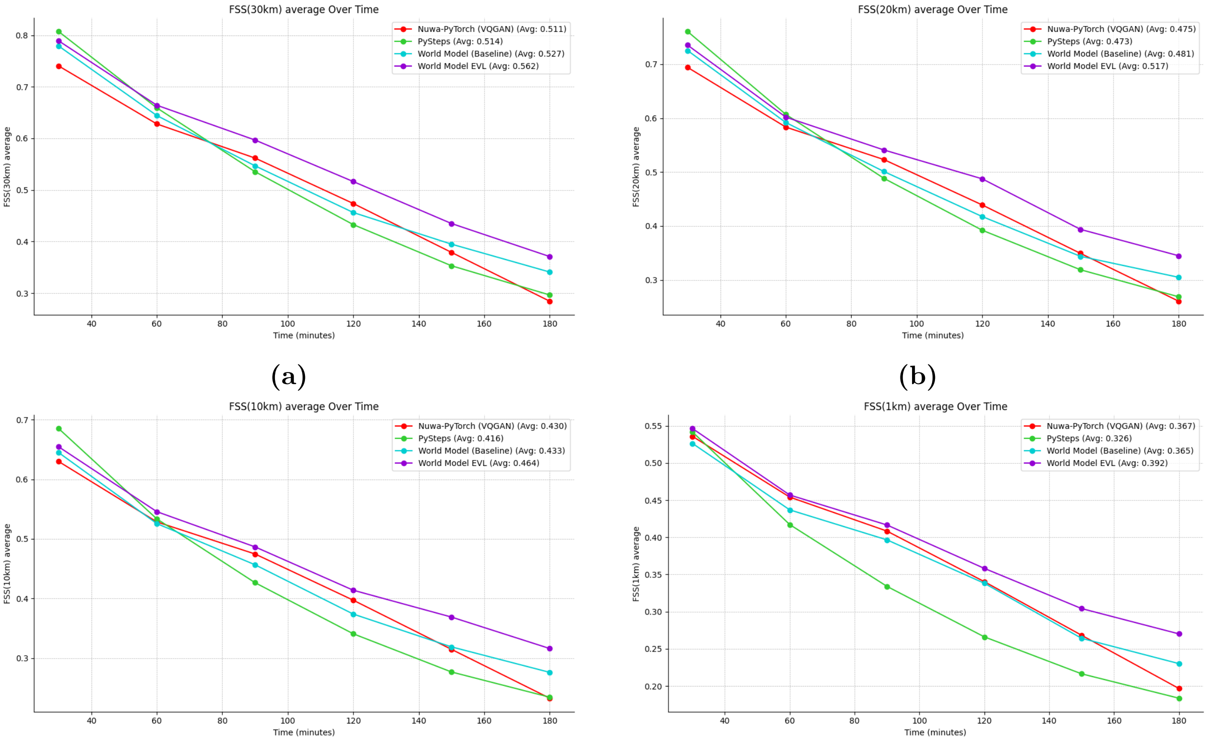
(a) **(b)**

Figure 5.3: 3-hour Nowcasting performance verification on the whole Netherlands region: Categorical Scores - CSI and FAR for a, b) 8mm. Relation between lead time and metric scores, with 3-hour averaged scores shown in the legend.

Figure 5.3 shows the CSI and FAR metrics for heavy rainfall in the entire Netherlands region. Sub-figure 5.3a shows the CSI metrics for the four models with 8mm as the threshold. It can be observed that PySTEPS has the highest CSI average when compared with the other models. However, it can also be seen that **World Model-EVL** has an overall consistent behavior over the whole horizon when compared with PySTEPS. **World Model-EVL** and **Nuwa-PyTorch (VQGAN)** have almost similar CSI average metrics over the whole horizon. However, based on the trend of the graphs of the two models, it can be observed that **World Model-EVL** displays better performance with increment in lead time over **Nuwa-PyTorch (VQGAN)**.

From sub-figure 5.3b, it can be observed that **World Model-EVL** has the lowest FAR average when compared with all the other models. This shows that **World Model-EVL** is more accurate in predicting heavier precipitation events. This is also in alliance with the fact that **World Model-EVL** has been made more sensitive to predicting heavier rainfall events (flood detection) with the incorporation of the EVL (Extreme Value Loss) function in the auto-regressive transformer part of the model.

Therefore, from the two above-mentioned figures: 5.2 and 5.3, it can be concluded that **World Model-EVL** shows comparable performance with the state-of-the-art benchmark PySTEPS in predicting lighter as well as heavier rainfall events in the whole Netherlands region. However, these plots have been calculated on the whole Netherlands region which has some additional noise in the prediction maps when compared with the ground-truth maps, and to have a better understanding of the performance of the models in detecting extreme events, catchment-level analysis has been performed and analyzed in the subsequent sections of this chapter.



(a) **(b)**

(c) **(d)**

Figure 5.4: 3-hour Nowcasting performance verification on the whole Netherlands region: FSS Scores for a) 30km, b) 20km, c) 10km, d) 1km. Relation between lead time and metric scores, with 3-hour averaged scores shown in the legend.

In the above-mentioned figure, FSS scores have been plotted for different length scales (1km, 10km, 20km, 30km) of the four models, respectively. Ideally, a larger length scale results in a larger FSS score which is evident from sub-figure 5.4a. Intuitively, it means that when we upscale the predictions to a coarser resolution, there would be less error in predicting precipitation field location. From figure 5.4, it can be concluded that **World Model-EVL** shows the best performance when compared with the other models on all the length scales. The FSS metrics for different length scales can also be used to provide intuitions on catchment-level analysis since, the length scales (1km, 10km, 20km, 30km) can be used to approximate different catchment areas. For instance, one of the catchments has roughly an area of $30km \times 30km$. Therefore, if a model has a higher FSS metric at 30 km length scale then the model must have a more accurate catchment-level prediction for a catchment that is roughly $30km \times 30km$.

Conclusion

From the above figures, it can be concluded that **World Model-EVL** shows comparable performance with respect to the other benchmark models presented in this analysis. With regards to continuous metrics such as PCC, MAE, FSS the model displays an overall better performance. In the context of categorical metrics such as CSI, FAR the model shows comparable performance with respect to PySTEPS and **Nuwa-PyTorch (VQGAN)**.

In figure 5.2, it can be observed that **World Model-EVL** has a higher CSI average metric for light rainfall (thresholds 1mm, 2mm) when compared with **Nuwa-PyTorch (VQGAN)** but shows comparable performance on the detection of heavy rainfall (threshold 8mm). Moreover, **World Model-EVL** shows an overall lower FAR average for both light and heavy rainfall detection in comparison to **Nuwa-PyTorch (VQGAN)**. This shows the significance of the implementation of the EVL loss function in the auto-regressive transformers of both the models. However, the **World Model-EVL** has an additional transformer (similar structure to a Vision Transformer) to classify the tokens as extreme or non-extreme on-the-fly while, training the auto-regressive transformer with the EVL loss function.

The baseline - **World Model** does not have any additional deep neural network for the classification of extreme/non-extreme tokens or the implementation of the EVL loss function which is evident from the lower CSI average values for all the thresholds (1mm, 2mm, 8mm) when compared with PySTEPS and **Nuwa-PyTorch (VQGAN)**. However, the model also has a lower FAR for all the thresholds when compared to **Nuwa-PyTorch (VQGAN)** suggesting that the VQVAE responsible for capturing the precipitation intensities into tokens is more powerful than the VQVAE of **Nuwa-PyTorch (VQGAN)**.

5.1.2 Effect of ensemble and post-processing technique

In this section, the continuous as well as the categorical metrics are calculated on the prediction frames for an ensemble of 5 for the deep generative models (**Nuwa-PyTorch (VQGAN)**, **World Model-EVL** and, **World Model** and an ensemble of 20 for PySTEPS since it is a Numerical Weather Prediction (NWP) model. The number 5 has been chosen as an ensemble number because the authors of [6] the **Nuwa-PyTorch (VQGAN)** model use this as the ensemble number. Hence, to be consistent in the comparison of metrics this number has been chosen. Also, more ensembles mean more amount of generation time so to maintain the trade-off between generation time and the ensemble number, Bi et al. [6] chose an ensemble of 5 predictions.

Metrics/Models	**PySTEPs**	**Nuwa-PyTorch**	**World Model(EVL)**	**World Model**
PCC (↑)	0.219	0.210	**0.253**	0.241
MAE (↓)	0.798	0.926	**0.714**	0.752
CSI (1mm) (↑)	0.250	0.262	**0.267**	0.254
CSI (8mm) (↑)	0.008	0.006	**0.009**	0.006
FAR (1mm) (↓)	0.617	0.618	0.587	**0.579**
FAR (8mm) (↓)	0.592	**0.399**	0.502	0.513
FSS (1km) (↑)	0.375	0.394	**0.432**	0.414
FSS (10km) (↑)	0.467	0.456	**0.493**	0.463
FSS (20km) (↑)	0.522	0.498	**0.534**	0.508

Table 5.1: Summary of the averaged precipitation scores over all the lead times for different models

In the table mentioned above 5.1, the average metrics over the whole horizon (all the lead times) have been shown for the four models, respectively. For the categorical metrics (CSI & FAR), 1mm threshold and 8mm threshold metrics have been shown since light and heavy rainfall detection can be categorized by these two thresholds. As for the continuous metrics, MAE, PCC, and FSS (for different length scales) have been shown. The highest value for each metric has been highlighted in **Bold** while the second-highest value for each metric has been underlined.

Averaging (Ensemble) assists in lowering the FAR metrics (averaged over all the lead times) of the prediction frames since the number of false positives reduces when predictions are averaged. However, it also reduces CSI, especially for the 2mm and the 8mm thresholds. This decrease could probably be explained by the lighter overall rainfall intensity in the averaging results. The continuous metrics such as MAE, PCC, and FSS (for all length scales) show improvement as well.

To tackle this, a post-processing technique (4.2.2) is introduced which helps in the increment of CSI metrics. This behavior is observed since, the post-processing helps in up-scaling the precipitation intensities of the pixels, thus, improving the detection of true positives for heavy rainfall (8mm threshold). However, it also worsens the other metrics such as FAR, MAE, and PCC but since these two techniques are applied together, an overall increment in all the metrics is observed for all the models in this experiment.

Conclusion

From table 5.1, it can be concluded that **World Model-EVL** has the overall best performance when compared with all the other models. The model displays better metrics in terms of both continuous (PCC, MAE, FSS) as well as categorical (CSI, FAR) metrics. The model displays better performance than PySTEPS thus, validating the usefulness of the ensemble as well as the post-processing technique in nowcasting tasks. In terms of CSI metric of 1mm threshold, **Nuwa-PyTorch (VQGAN)** shows comparable performance with regard to **World Model-EVL**, highlighting the significance of the incorporation of the EVL loss function in its auto-regressive transformer. Moreover, **World Model-EVL** also showcases a lower FAR metric for 1mm threshold, when compared with **Nuwa-PyTorch (VQGAN)**. Therefore, based on this trade-off, it can be concluded that the overall performance in detecting light rainfall (1mm threshold) is better for **World Model-EVL**.

Similarly, in the detection of heavy rainfall (8mm threshold) in the whole region of Netherlands, **World Model-EVL** displays the best CSI metric when compared with all the other models especially when compared with **Nuwa-PyTorch (VQGAN)** (there is a 50% increment in the CSI 8mm metric). However, **Nuwa-PyTorch (VQGAN)** also shows the lowest FAR metric for the 8mm threshold, followed by **World Model-EVL**. The comparison suggests that while both models — **Nuwa-PyTorch (VQGAN)** and **World Model-EVL** — show nearly equivalent effectiveness in identifying heavy rainfall throughout the whole region of Netherlands, **World Model-EVL** model stands out. It demonstrates a notably higher increase in the CSI (while maintaining a reasonable FAR value), indicating a better ability to detect heavy rainfall events. Moreover, PySTEPS also displays comparable performance in the detection of heavy rainfall (8mm) but at the trade-off of a comparatively high FAR metric.

It can also be observed that the baseline model **World Model** shows comparable performance with respect to PySTEPS and **Nuwa-PyTorch (VQGAN)** in the detection of both heavy (8mm) and light (1mm) rainfall events. Moreover, the model also shows better performance with respect to continuous metrics such as MAE, and PCC when compared with PySTEPS and **Nuwa-PyTorch (VQGAN)**. This proves that even the baseline model (without the application of the EVL loss function and the binary classifier) is suitable for precipitation nowcasting tasks.

5.2 Extreme event detection in catchment regions

In this section, the identification of extreme weather events within specific catchments in the entire Netherlands region is assessed and analyzed. To do this, the relevant sections of the predicted precipitation maps, generated by various models, are isolated to examine just the catchment areas in question. An 'extreme event' in this context, is characterized by the average amount of rainfall collected over a 3-hour period within that specific catchment. The thresholds are the top 1% highest average precipitation accumulation (based on the KNMI-RT dataset) as well as the top 5% highest average precipitation accumulation over the respective catchments. Based on this, the thresholds for each catchment are then calculated.

The detection ability of the models has been assessed in two ways:

- Firstly, each catchment area is assigned a specific extreme rainfall threshold that has been previously calculated. The extreme rainfall thresholds are first calculated for the top 1% and the top 5% of the events. Then, the models' performance at this threshold is measured using four key categorical metrics: Hit Rate (HR), False Alarms (FA), False Alarm Ratio (FAR), and Critical Success Index (CSI) whose formulae have been described in the respective tables below.
- Secondly, to get a broader view of the models' abilities to detect extreme events, a range of different extreme rainfall thresholds is applied uniformly across all the predictions of the catchment areas, while keeping the threshold on the ground truth constant. The models' performance are then compared using a Receiver Operating Characteristic (ROC) curve and a Precision-Recall curve, which visually represents their ability to detect extreme events under these varied thresholds. In this analysis, it is assumed that every catchment area is subject to the same threshold for what constitutes extreme rainfall.

The equations involved in calculating the 3-hour averaged precipitation accumulation for the ground truth catchment areas as well as the predicted catchment areas are as given below:

$$\begin{aligned} X_{pre} &= \left(X_{pre}^{T+30} + X_{pre}^{T+60} + \ldots + X_{pre}^{T+180}\right) * \frac{1}{6} * 3 \\ X_{obs} &= \left(X_{obs}^{T+30} + X_{obs}^{T+60} + \ldots + X_{obs}^{T+180}\right) * \frac{1}{6} * 3 \end{aligned} \tag{5.1}$$

In the above equation 5.1, X_{pre} represents the precipitation accumulation estimated from the prediction, whereas X_{obs} represents the ground-truth precipitation accumulation from the KNMI - Real Time (RT) dataset. The unit for both, X_{pre} and X_{obs} is mm/3hr. The experiment uses 357 events across the entire Netherlands region and 3,927 events at the catchment level, occurring between 2019 and 2021. Every nationwide event includes one or more extreme events at the catchment level.

5.2.1 Fixed Threshold Evaluation

For the fixed threshold evaluation, the top 1% highest average precipitation levels, based on all rainfall events within each of the twelve catchment areas from 2008 to 2014, have been established as the extreme rainfall thresholds for these catchments. The magnitude of this threshold is approximately 5mm/3hr. The performance of the models has been assessed using four categorical metrics: Hit Rate (HR), False Alarm rate (FA), False Alarm Ratio (FAR) and, Critical Success Index (CSI). The results obtained for the four models have been displayed below in the table 5.2 with the formulae for each metrics used for this analysis (where, H: True Positive, M: False Negative, F: False Positive, R: True Negative). The best values are highlighted in **Bold** while, the second best values are underlined.

From the below-mentioned 5.2 table, it can be observed that **World Model** and **World Model-EVL** models show overall better performance than PySTEPS and **Nuwa-PyTorch (VQGAN)**. **World Model-EVL** model has a higher HR as well as, a low FAR when compared with **Nuwa-PyTorch (VQGAN)**. This shows that the **World Model-EVL** displays better performance in detecting heavy rainfall events (since, only top 1% largest catchment average precipitation has been chosen as the threshold for this evaluation) when compared with **Nuwa-PyTorch (VQGAN)**. Also, the baseline model **World Model**, shows a similar CSI metric when compared with **Nuwa-PyTorch (VQGAN)** which shows that the model also has a promising ability in the detection of heavy rainfall events in catchment areas.

Models/Metrics	HR = H/(H+M)↑	FA = F/(R+F)↓	FAR = F/(H+F)↓	CSI = H/(H+M+F)↑
PySTEPS	0.3838	0.0965	0.4812	0.2830
Nuwa-PyTorch	0.3959	0.1205	0.5288	0.2941
World Model	0.4006	0.0806	0.4156	0.2998
Weather Model-EVL	**0.4209**	**0.0733**	**0.3903**	**0.3109**

Table 5.2: Summary of the 1% extreme event detection performance of different models (Catchment-level evaluation, RT dataset) 5mm/3h.

Conclusion

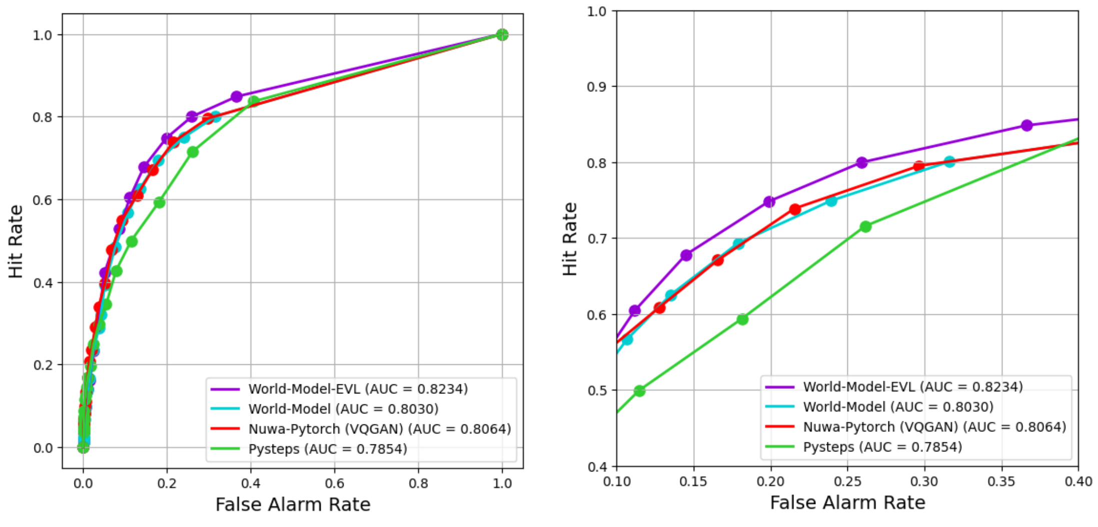
From the above table 5.2, it can be concluded that the proposed model **World Model-EVL** exhibits better performance in the detection of heavy rainfall (5mm/3hr) when compared with PySTEPS and **Nuwa-PyTorch (VQGAN)**. The increment in CSI metric proves the model's effectiveness in detecting extreme events using a highly imbalanced dataset (the number of extreme events is significantly less than the number of normal events) such as the one used in this experiment.

In comparison to **Nuwa-PyTorch (VQGAN)**, **World Model-EVL** shows better performance with regard to all the metrics. This proves that the implementation of the Binary classifier (for the classification of extreme and non-extreme tokens) along with the EVL loss function in the autoregressive transformer part of the model is more effective than just assuming the total number of extreme tokens in the discrete latent space, generated by the VQVAE part of the model (the main approach of the **Nuwa-PyTorch (VQGAN)** model).

The model **World Model-EVL** also achieves better metrics for the detection of heavy rainfall in the catchment regions in comparison to **World Model**. This proves the effectiveness of the application of the EVL loss function to bias the model towards predicting extreme tokens while generation. Also, another advantage of using the EVL loss function is that it utilizes Extreme Value Theory (EVT) to estimate the weights in the EVL loss function (as shown in equation (2.30)), working under the assumption that the extreme values in the dataset follow a heavy-tailed distribution, specifically a Type II GEV distribution. This allows the **World Model-EVL** to better account for and represent the extreme values in the data.

5.2.2 Evaluation of overall extreme event detection ability

To gain a clearer insight into the capability of the respective models to identify extreme weather events at various levels, an additional experiment is performed. Instead of using a fixed threshold for the predictions of each catchment, a series of varying thresholds are applied to all the catchment predictions. For the ground truth data, the same threshold is maintained (the extreme threshold for each catchment is set to the top 1% highest average precipitation levels i.e. 5mm/3hr). For this analysis, a common set of descending thresholds are established for the predictive data at 10, 9.5, 9, 8.5, 8, 7.5, 7, 6.5, 6, 5.5, 5, 4.5, 4, 3.5, 3, 2.5, 2, 1.5, 1 and, 0.5 mm/3hr (20 data points). The results of this experiment are illustrated through a Receiver Operating Characteristic (ROC) curve and a Precision-Recall curve in the below-mentioned figures 5.5 and 5.6.

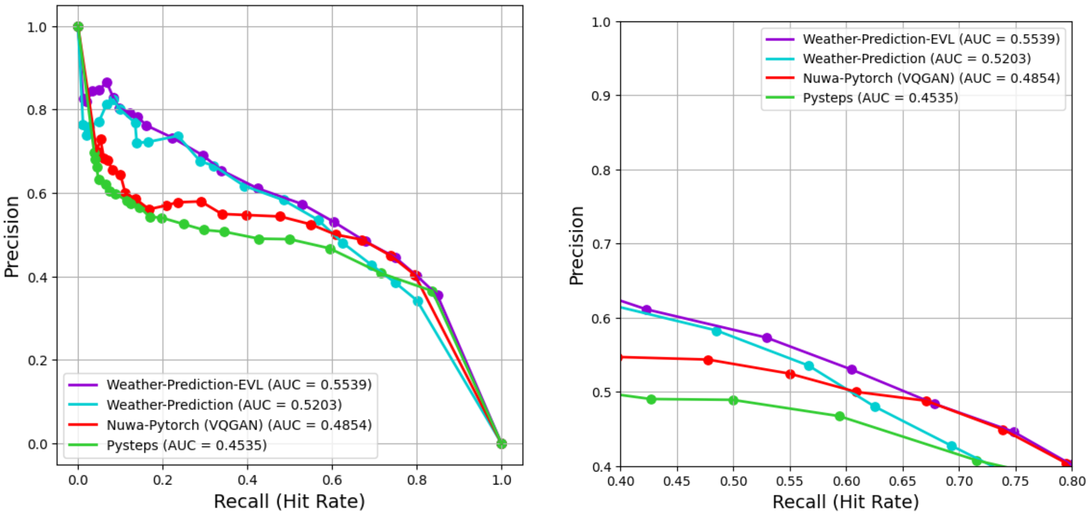


(a) **(b)**

Figure 5.5: (a) The complete ROC curve for 3-hour extreme event detection, the points on the curve (from left to right) represent thresholds from 10mm to 0.5mm. (b) Cropping of the ROC curve by limiting the hit rate to be higher than 0.4 and false alarm rate lower than 0.4.

By comparing the Area under Curve (AUC) in the below-mentioned sub-figure 5.5a, it can be observed that the model **World Model-EVL** outperforms all the other models that are part of this experiment. Moreover, the baseline **World Model** displays a similar performance when compared with **Nuwa-PyTorch (VQGAN)**. This shows that even though **World Model** model does not have the additional implementation of the EVL loss function, the overall choice of the VQVAE as well as, the auto-regressive transformer makes it powerful enough to have a comparable extreme event detection performance with respect to **Nuwa-PyTorch (VQGAN)**.

In terms of the complete curve, it is difficult to assess the detection performance of the different models so the HR has been limited from 0.4 - 1 and the FAR from 0.1 - 0.4 for a better analysis and is shown in sub-figure 5.5b. From this figure, it can be observed that **World Model-EVL** has a higher HR metric when compared with all the other models. Also, within this limited range of FAR, the **World Model-EVL** has Hit rate (HR) values above 0.6 for all the data points (5 data points so for 5 thresholds out of the 20 thresholds mentioned above) in the plot when compared with all the other models.



(a) (b)

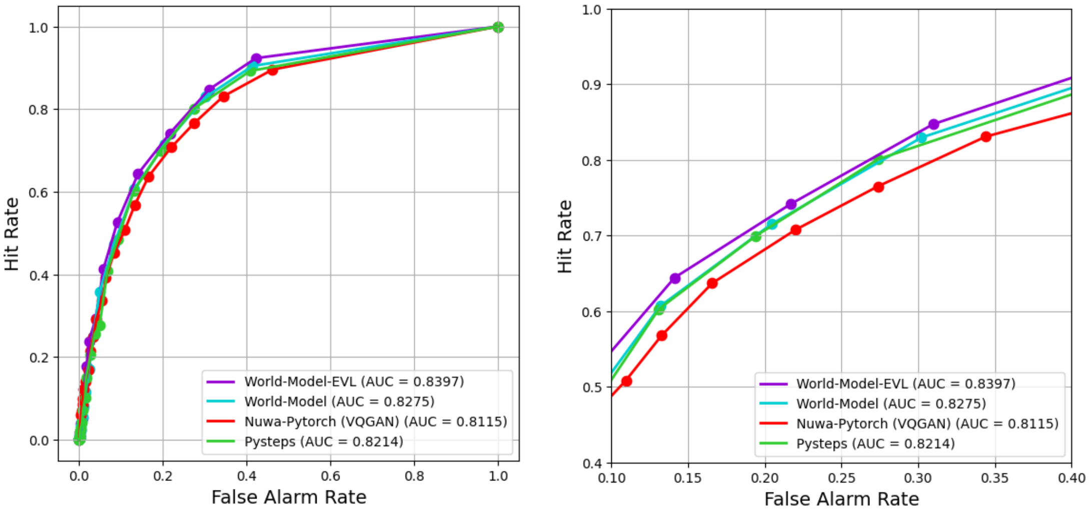
Figure 5.6: (a) The complete Precision-Recall curve for 3-hour extreme event detection, the points on the curve (from left to right) represent thresholds from 10mm to 0.5mm. (b) Cropping of the Precision-Recall curve by limiting both precision as well as recall to be higher than 0.4.

In the above analysis with the ROC curves, the False Alarm Rate (FAR) metric gets normalized by the number of True Negatives (R) which is a huge number. Therefore, for a more meaningful analysis of the extreme event detection performance of the models, Precision-Recall Curve has also been plotted and shown in the below-mentioned figure 5.6. For this analysis as well, a common set of descending thresholds are established for the predictions at 10, 9.5, 9, 8.5, 8, 7.5, 7, 6.5, 6, 5.5, 5, 4.5, 4, 3.5, 3, 2.5, 2, 1.5, 1 and, 0.5 mm/3hr (20 data points) while the threshold for the ground-truth data has been set to 5mm/3hr (top 1%).

From the above sub-figure 5.6a, it can be observed that **World Model-EVL** has the highest AUC when compared with all the other models. To have a better understanding of the performance of the models, the recall (Hit Rate) has been limited to the range 0.4 - 0.8 while, precision scores in the range 0.4 - 1, as shown in sub-figure 5.6b. It can be observed that with the increment in the recall scores (approximately, around 0.65 and higher) **World Model-EVL** and **Nuwa-PyTorch (VQGAN)** have almost similar Precision scores. However, on the basis of the overall AUC value for all the models, it can be concluded that **World Model-EVL** has the best performance amongst all the models in this analysis.

5.2.3 Evaluation of overall moderate rainfall detection ability

The performance of the different models has also been analyzed on the top 5% highest average precipitation levels i.e. 2mm/3hr. However, this threshold cannot be considered as an extreme threshold and thus, shows the performance of the models in the detection of moderate rainfall. For this analysis as well, a common set of descending thresholds are established for the predictions, similar to the ones mentioned in the above sub-section 5.2.2. The analysis is done using an ROC curve (figure 5.7) as well as a Precision-Recall curve as depicted below in the subsequent parts of this section.



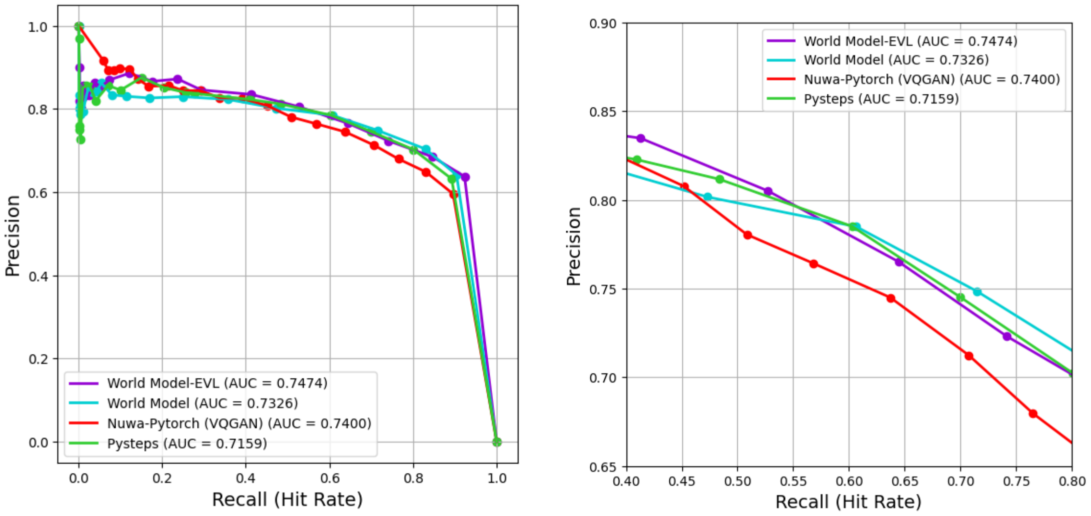
(a) **(b)**

Figure 5.7: (a) The complete ROC curve for 3-hour moderate rainfall event detection, the points on the curve (from left to right) represent thresholds from 10mm to 0.5mm. (b) Cropping of the ROC curve by limiting the hit rate to be higher than 0.4 and false alarm rate lower than 0.4.

From the below-mentioned sub-figure 5.7a, it can be observed that all the models have almost similar performance in the detection of moderate rainfall. Based on the magnitude of the AUC of the different models, **World Model-EVL** displays the best performance. Since the plots of each model overlap significantly, sub-figure 5.7b shows the performance of the models in a limited range of Hit Rate of 0.4-1 and False Alarm Rate of 0.1-0.4. From this figure, it can be observed that PySTEPS and **World Model** display almost similar performance in their detection abilities in this range. Therefore, it can be concluded that even though PySTEPS shows comparatively lower performance in the detection of extreme events, it displays promising performance in the detection of moderate rainfall events.

From sub-figure 5.7b it can also be observed that **Nuwa-PyTorch (VQGAN)** has comparatively lower performance in the detection of moderate rainfall events when compared with the **World Model-EVL**. This shows the robustness of the additional implementation of the Binary Classifier for the classification of the extreme and non-extreme tokens along with the incorporation of the EVL loss function in the Auto-regressive Transformer of the **World Model-EVL**.

In the below-mentioned figure 5.8, precision-recall curves have been plotted for the different models in a similar fashion as described in sub-section 5.2.2 but with the threshold of 2mm/3hr for the ground-truth data. It can be observed from sub-figure 5.8a that AUC for **World Model-EVL** is the highest amongst all the other models in comparison. However, to have a better understanding of the performance of the different models the precision scores have been limited to 0.65-0.9 while the recall scores to 0.4-0.8 as shown in sub-figure 5.8b. It can be observed that with the increment in Recall scores (0.65 and above), the precision of **World Model-EVL** reduces when compared with **World Model** and PySTEPS. This shows that both **World Model** and PySTEPS have comparable performance in the detection of moderate rainfall events when compared with **World Model-EVL**.



(a) **(b)**

Figure 5.8: (a) The complete Precision-Recall curve for 3-hour moderate rainfall event detection, the points on the curve (from left to right) represent thresholds from 10mm to 0.5mm. (b) Cropping of the Precision-Recall curve by limiting both precision to be higher than 0.65 and recall to be higher than 0.40.

5.2.4 Evaluation of extreme events in specific catchments

In this section, the ROC and Precision-Recall Curves for two specific catchments have been shown and analyzed for the different models, part of this thesis. The ROC and the Precision-Recall curves have been constructed in the same way as mentioned above in the above-section 5.2.2 i.e. to have the same threshold for the ground-truth data (threshold of 5mm/3hr) while having a common set of descending thresholds (10, 9.5, 9, 8.5, 8, 7.5, 7, 6.5, 6, 5.5, 5, 4.5, 4, 3.5, 3, 2.5, 2, 1.5, 1 and, 0.5 mm/3hr) for the predictive data.

The catchments chosen for this evaluation are **Regge** and **Delfland**. Regge is the largest catchment in the Netherlands region spanning a region of $957km^2$. Hence, it is necessary to analyze the detection ability of the different models in this region. The second catchment that was chosen for this analysis is Delfland which is the third largest catchment (out of the 12 catchments), spanning a region of approximately $379km^2$.

Catchment: Regge

In this section, the ROC and the Precision-Recall curve for the catchment have been shown in figures 5.9 and 5.10, respectively. From sub-figure 5.9a, it can be observed that **World Model(EVL)** has the highest AUC amongst all the models. However, the analysis with a limited range of Hit Rate and False Alarm rate in sub-figure 5.9b shows that the baseline **World Model** has almost similar performance with respect to **World Model-EVL** (the models display similar performance till the Hit Rate of 0.77).

From the below-mentioned figure 5.9, it can be concluded that all the models show similar performance in the detection of extreme rainfall events in Regge especially when analyzed in the limited Hit Rate and False Alarm rate range.

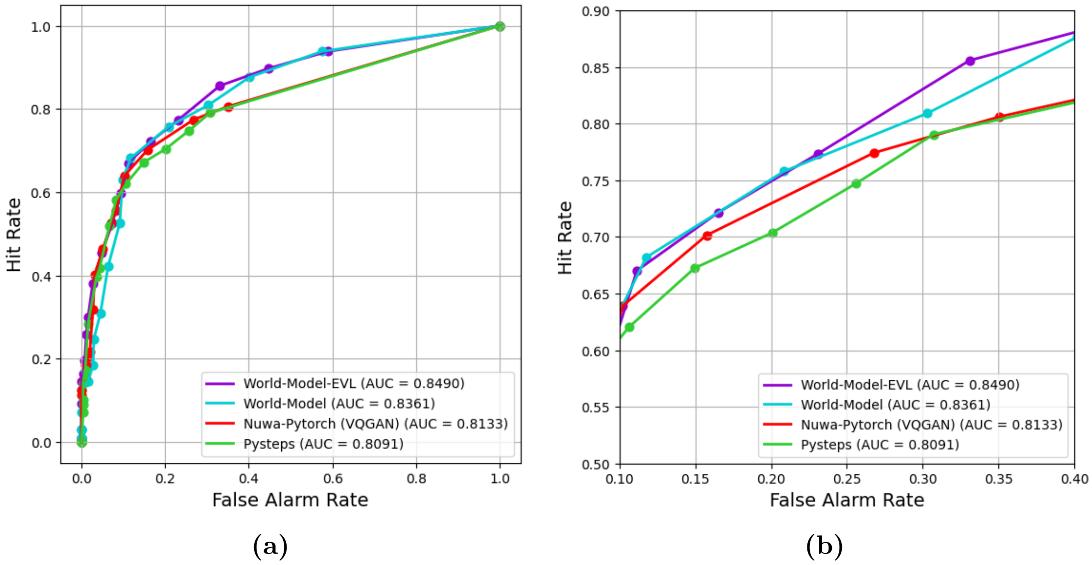


Figure 5.9: (a) The complete ROC curve for 3-hour extreme rainfall event detection for catchment Regge, the points on the curve (from left to right) represent thresholds from 10mm to 0.5mm. (b) Cropping of the ROC curve by limiting the hit rate to be higher than 0.5 and false alarm rate lower than 0.4.

Based on the Precision-Recall Curve shown in figure 5.10, it can be observed that **World Model-EVL** shows an overall better performance when compared with the other models. The baseline **World Model** also shows a comparable performance with respect to **Nuwa-PyTorch (VQGAN)** and PySTEPS.

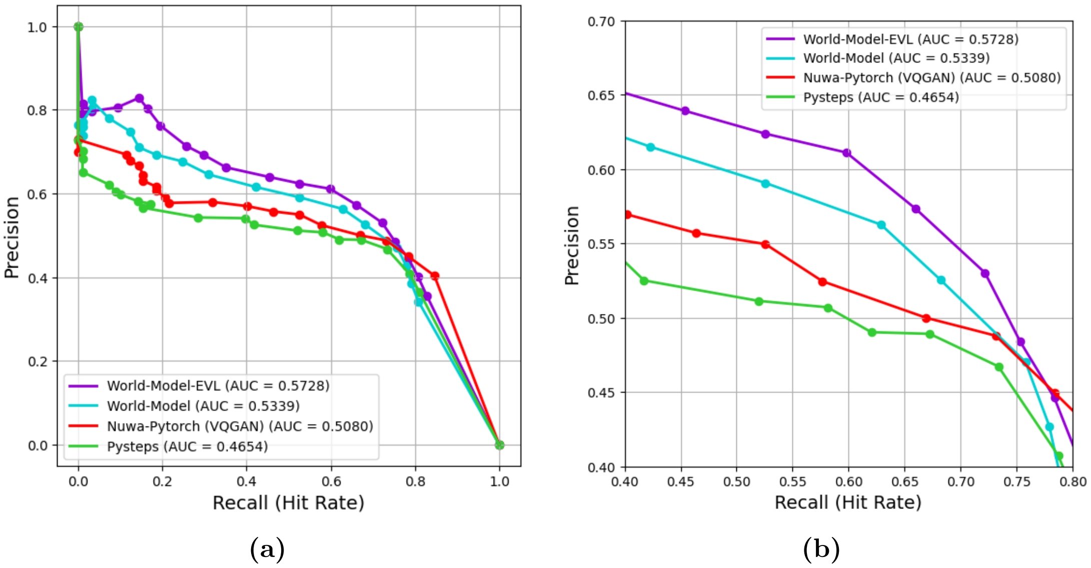


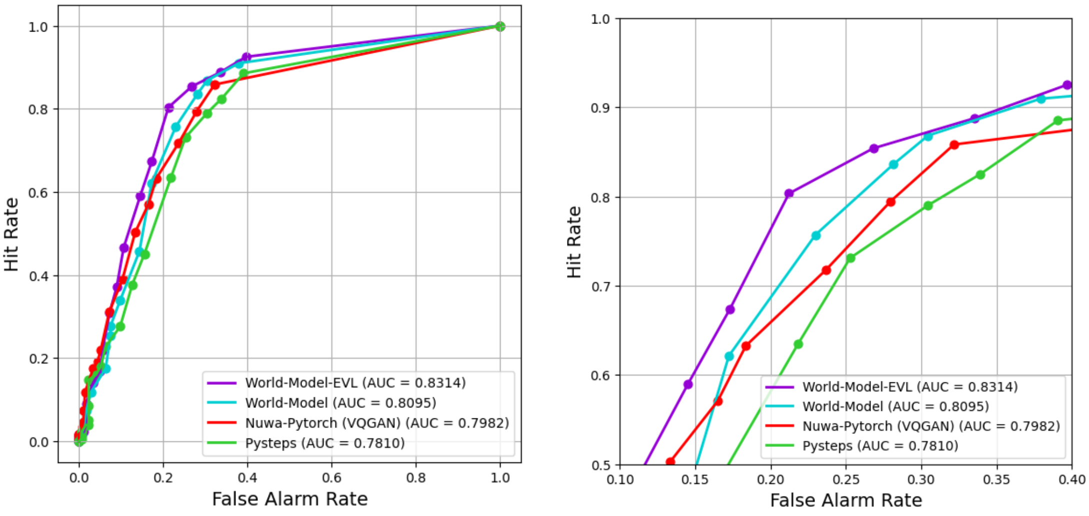
Figure 5.10: (a) The complete Precision-Recall curve for 3-hour extreme rainfall event detection, the points on the curve (from left to right) represent thresholds from 10mm to 0.5mm. (b) Cropping of the Precision-Recall curve by limiting both precision as well as recall to be higher than 0.4.

However, it can also be observed from figure 5.10b that with the increment in Recall (0.7 onwards), the performance of **World Model** decreases whereas **Nuwa-PyTorch (VQGAN)** and **World Model-EVL** display almost similar behavior.

Catchment: Delfland

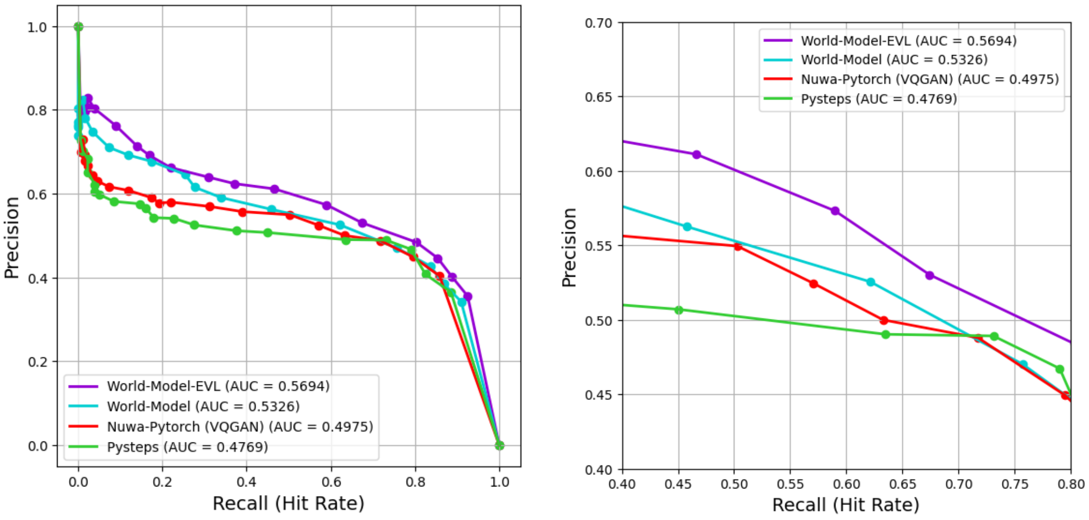
In this section, the ROC curve and the Precision-Recall Curve have been shown for the catchment Delfland in figures 5.11 and 5.12 for the top 1% of extreme events. From figure 5.11 it can be observed that **World Model-EVL** shows the overall best performance when compared with the other models based on AUC. Moreover, it can also be concluded that the deep learning models part of this thesis display almost similar performance based on figure 5.11b.

In the below-mentioned figure 5.12, the precision-recall curve has been plotted for the four respective models for the top 1% of extreme events. Based on the AUC, it can be observed that **World Model-EVL** displays an overall better performance when compared with the other models. However, based on figure 5.12b, it can be observed that within the limited Recall range **World Model-EVL** has a higher precision but with the increment in recall scores, the performance of the model decreases.



(a) **(b)**

Figure 5.11: (a) The complete ROC curve for 3-hour extreme rainfall event detection for catchment Delfland, the points on the curve (from left to right) represent thresholds from 10mm to 0.5mm. (b) Cropping of the ROC curve by limiting the hit rate to be higher than 0.4 and false alarm rate lower than 0.4.



(a) **(b)**

Figure 5.12: (a) The complete Precision-Recall curve for 3-hour extreme rainfall event detection, the points on the curve (from left to right) represent thresholds from 10mm to 0.5mm. (b) Cropping of the Precision-Recall curve by limiting both precision as well as recall to be higher than 0.4.

Therefore, based on the analysis shown by the above figures, it can be concluded that **World Model-EVL** displays an overall higher performance in the detection of extreme events when compared with the other models. However, **Nuwa-PyTorch (VQGAN)** and the baseline **World Model** also show comparable performance in the detection of extreme events in the catchment areas when compared with **World Model-EVL**.

Conclusion and Future Scope 6

6.1 Conclusion

This section concludes the main contributions of this thesis project and the answer to the research objective mentioned in Chapter 1. In this thesis work, a transformer-based deep generative model has been proposed for extreme precipitation nowcasting. Based on the literature research, transformer-based generative models have better performance in capturing long-term dependencies between features when compared to existing deep neural networks based on ConvLSTM. However, the dataset utilized in this thesis consists of high-resolution images which can raise computational constraints if the transformer itself is trained on it (the attention mechanism in transformers has quadratic complexity, thus making it computationally infeasible). Therefore, a VQ-VAE is incorporated as well, inspired by the research of Esser et. al [29], Bi et. al [6] which constructs a low-dimensional discrete representation of the inputs. The transformer gets trained on these discrete-level representations of the images in an autoregressive fashion which are then decoded back to the original spatial resolutions of the radar precipitation maps with the help of the decoder of the VQ-VAE.

Furthermore, the goal of this thesis work is to also develop a model that shows comparable nowcasting performance in the entire region of the Netherlands as well as extreme precipitation detection in the corresponding catchment regions, detailed in Chapter 3. However, based on the analysis of the KNMI RT-radar dataset (table 3.5 and table 3.8) it can be observed that the distribution of the data is highly imbalanced with regard to extreme precipitation data. The occurrence of light rainfall intensities ($X \leq 0.1\, mm/h$) comprises of approximately 90% of the precipitation intensities when analyzed for the whole Netherlands region. However, an analysis on the entire Netherlands region is not enough since a single pixel displaying extreme precipitation at a single time point might not represent an extreme precipitation event in a real-world scenario. Therefore, a catchment-averaged rainfall accumulation analysis has been performed to define extreme events. This analysis helps in covering both the spatial as well as temporal aspects of the occurrence of an extreme event.

The dataset's significant imbalance leads to an uneven distribution of discrete tokens in the latent space of the VQ-VAE part of the model, which also affects the input data for the autoregressive Transformer. Essentially, training the Transformer involves tackling a multi-class classification challenge by employing a cross-entropy loss to categorize the current tokens into one of the 1024 tokens in the codebook of the VQ-VAE. This kind of imbalance may result in underfitting, adversely affecting the model's performance in precipitation nowcasting and extreme event detection. Therefore, to mitigate this issue, the EVL loss function has been implemented in the autoregressive transformer along with a Binary classifier that helps in the classification of the tokens as extreme or non-extreme based on area-averaged precipitation of 5mm over the ground-truth radar map. This helps the autoregressive transformer to handle the uneven distribution of the discrete tokens in the latent space and generate tokens that encode extreme rainfall over an area of $16km \times 16km$ (since the input radar map has a spatial resolution of $128km \times 128km$ and every map is encoded to 8×8 discrete tokens by the VQ-VAE).

Based on the experiments and results shown in Chapter 5, it can be concluded that the proposed model **World Model-EVL** displays comparatively better overall nowcasting performance on the entire Netherlands region with respect to other models. The model shows significant improvement in the extreme event detection as well, when compared with benchmark models such as PySTEPS and **Nuwa-PyTorch (VQGAN)**. Moreover, it also shows the application of the EVL loss function in a more robust way when compared with **Nuwa-PyTorch (VQGAN)** in which the author assumes a fixed empirical distribution of the discrete latent representation of extreme precipitation (i.e. the distribution of the extreme tokens).

6.2 Future Scope

One of the main difficulties faced in this project is the scarcity of extreme precipitation maps. This hinders the training of the VQ-VAE part of the model as it is unable to learn an adequate amount of discrete latent representations (tokens) of radar images that display extreme precipitation. Consequently, during prediction, the autoregressive transformer faces difficulty in generating tokens that capture extreme rainfall. Even though the incorporation of the additional binary classifier and the EVL loss function helps in better prediction of precipitation maps that encompass extreme precipitation, it would be beneficial to perform data augmentation to increase the number of input radar images that showcase heavy or extreme rainfall. However, data augmentation should be only implemented for radar images that display extreme rainfall intensities since the number of such images is quite low. This would help the VQ-VAE to learn a better representation of the extreme precipitation images in the discrete latent space which in turn would help the autoregressive transformer to predict more extreme tokens (since this would increase the latent distribution of extreme tokens).

Moreover, research by Bi et. al [6], Esser et. al [29], Yan et. al [32] [39] show that incorporating a spatial discriminator with the VQ-VAE helps in learning a better discrete latent representation of high-resolution images (such as the images in our dataset). This occurs because of the additional adversarial loss incorporation with the VQ-VAE loss shown in equation (4.2). The adversarial training process (typically, a *minmax* problem) provides a dynamic feedback mechanism to the VQ-VAE. The VQ-VAE is continually adjusted based on the discriminator's assessments, leading to iterative improvements in the generated outputs over the course of training.

Furthermore, VideoGPT proposed by Yan et. al [32] is also a VQVAE+autoregressive transformer-based deep learning model that works on videos rather than images. Therefore, the VQ-VAE in the case of VideoGPT has 3D convolutions in the VQ-VAE encoder, which helps extract temporal/depth features along with 2D spatial features. This can prove to be beneficial in learning better discrete latent representations of the images since, the input radar maps can inherently be treated as a sequence of images (i.e., a video sequence).

References

[1] R. Prudden, S. Adams, D. Kangin, *et al.*, "A review of radar-based nowcasting of precipitation and applicable machine learning techniques," 2020.

[2] R. O. Imhoff, C. C. Brauer, A. Overeem, A. H. Weerts, and R. Uijlenhoet, "Spatial and temporal evaluation of radar rainfall nowcasting techniques on 1,533 events," en, *Water Resour. Res.*, vol. 56, no. 8, Aug. 2020.

[3] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," 2015.

[4] V. Lebedev, V. Ivashkin, I. Rudenko, *et al.*, "Precipitation nowcasting with satellite imagery," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage AK USA: ACM, Jul. 2019.

[5] S. Ravuri, K. Lenc, M. Willson, *et al.*, "Skillful precipitation nowcasting using deep generative models of radar," 2021.

[6] H. Bi, M. Kyryliuk, Z. Wang, *et al.*, "Nowcasting of extreme precipitation using deep generative models," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece: IEEE, Jun. 2023.

[7] D. Ding, M. Zhang, X. Pan, M. Yang, and X. He, "Modeling extreme events in time series prediction," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage AK USA: ACM, Jul. 2019.

[8] U. Germann and I. Zawadzki, "Scale-dependence of the predictability of precipitation from continental radar images. part i: Description of the methodology," en, *Mon. Weather Rev.*, vol. 130, no. 12, pp. 2859–2873, Dec. 2002.

[9] A. W. Seed, C. E. Pierce, and K. Norman, "Formulation and evaluation of a scale decomposition-based stochastic precipitation nowcast scheme," en, *Water Resour. Res.*, vol. 49, no. 10, pp. 6624–6641, Oct. 2013.

[10] J. Liu, L. Xu, and N. Chen, "A spatiotemporal deep learning model ST-LSTM-SA for hourly rainfall forecasting using radar echo images," en, *J. Hydrol. (Amst.)*, vol. 609, no. 127748, p. 127 748, Jun. 2022.

[11] K. Trebing, T. Stanczyk, and S. Mehrkanoon, "SmaAt-UNet: Precipitation nowcasting using a small Attention-UNet architecture," 2020.

[12] A. Bojesomo, H. Al Marzouqi, and P. Liatsis, "Spatiotemporal swin-transformer network for short time weather forecasting," Sep. 2021.

[13] L. Xiang, J. Guan, J. Xiang, L. Zhang, and F. Zhang, "Spatiotemporal model based on transformer for bias correction and temporal downscaling of forecasts," *Front. Environ. Sci.*, vol. 10, Nov. 2022.

[14] T. Nguyen, J. Brandstetter, A. Kapoor, J. K. Gupta, and A. Grover, "ClimaX: A foundation model for weather and climate," 2023.

[15] P. Asadi, S. Engelke, and A. C. Davison, "Optimal regionalization of extreme value distributions for flood estimation," en, *J. Hydrol. (Amst.)*, vol. 556, pp. 182–193, Jan. 2018.

[16] Y. Boulaguiem, J. Zscheischler, E. Vignotto, K. van der Wiel, and S. Engelke, "Modeling and simulating spatial extremes by combining extreme value theory with generative adversarial networks," en, *Environ. Data Science*, vol. 1, no. e5, 2022.

[17] S. Coles, *An introduction to statistical modeling of extreme values* (Springer Series in Statistics). Springer-Verlag, 2001, ISBN: 1-85233-459-2.

[18] L. De Haan and A. Ferreira, *Extreme Value Theory* (Springer Series in Operations Research and Financial Engineering), en. Springer Science+Business Media, Jan. 2006.

[19] D. Levine, "Modeling tail behavior with extreme value theory," *Risk Management*, vol. 17, pp. 14–18, 2009.

[20] C. O. Omari, P. N. Mwita, and A. G. Waititu, "Using conditional extreme value theory to estimate value-at-risk for daily currency exchange rates," *J. Math. Fin.*, vol. 07, no. 04, pp. 846–870, 2017.

[21] S. Chen, N. Kalanat, S. Topp, *et al.*, "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*, Birmingham United Kingdom: ACM, Oct. 2023.

[22] R. Gençay and F. Selçuk, "Extreme value theory and Value-at-Risk: Relative performance in emerging markets," en, *Int. J. Forecast.*, vol. 20, no. 2, pp. 287–303, Apr. 2004.

[23] *KNMI - From Pulse to Product: Highlights of the digital-IF upgrade of the Dutch national radar network — knmi.nl*, https://www.knmi.nl/kennis-en-datacentrum/publicatie/from-pulse-to-product-highlights-of-the-digital-if-upgrade-of-the-dutch-national-radar-network.

[24] A. C. Best, "The size distribution of raindrops," en, *Q. J. R. Meteorol. Soc.*, vol. 76, no. 327, pp. 16–36, Jan. 1950.

[25] *Precipitation - radar/gauge 5 minute real-time accumulations over the Netherlands - KNMI Data Platform — dataplatform.knmi.nl*, https://dataplatform.knmi.nl/dataset/nl-rdr-data-rtcor-5m-1-0.

[26] *Precipitation - 5 minute precipitation accumulations from climatological gauge-adjusted radar dataset for The Netherlands (1 km) in KNMI HDF5 format - KNMI Data Platform — dataplatform.knmi.nl*, https://dataplatform.knmi.nl/dataset/rad-nl25-rac-mfbs-5min-2-0.

[27] Tim, *Heavy Tailed Distribution & Light Tailed Distribution: Definition & Examples — statisticshowto.com*, https://www.statisticshowto.com/heavy-tailed-distribution/.

[28] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," 2017.

[29] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," 2020.

[30] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," 2016.

[31] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," 2017.

[32] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas, "VideoGPT: Video generation using VQ-VAE and transformers," 2021.

[33] J. Lages, *Transformers KV Caching Explained — joaolages*, https://medium.com/@joaolages/kv-caching-explained-276520203249.

[34] G. Chen and W.-C. Wang, "Short-term precipitation prediction using deep learning," 2021.

[35] *Welcome to Lightning Fabric 2014; lightning 2.3.0 dev documentation — lightning.ai*, https://lightning.ai/docs/fabric/2.2.0/.

[36] *Automatic Mixed Precision for Deep Learning — developer.nvidia.com*, https://developer.nvidia.com/automatic-mixed-precision.

[37] *Weather Forecasting ... On-Line — wxonline.info*, https://www.wxonline.info/topics/verif2.html.

[38] J. Brownlee, *How to Use ROC Curves and Precision-Recall Curves for Classification in Python - MachineLearningMastery.com — machinelearningmastery.com*, https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/.

[39] W. Yan, D. Hafner, S. James, and P. Abbeel, “Temporally consistent transformers for video generation,” Oct. 2022. arXiv: 2210.02396 [cs.CV].

Appendix A

A.1 Representation of the Reconstructed frames

In this section, the Reconstructed images from the VQ-VAE of the **World Model (EVL)** have been displayed in figure A.1 as well as A.2 for two random batches of images from the testing dataset.

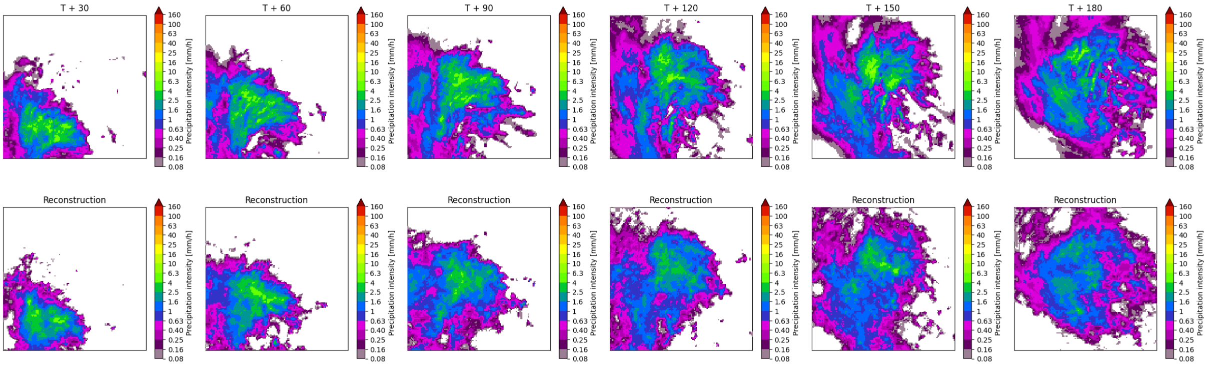


Figure A.1: Example of reconstruction of Precipitation fields by the VQ-VAE (Upper row: Original Radar Input image, Lower row: Reconstructed precipitation fields.

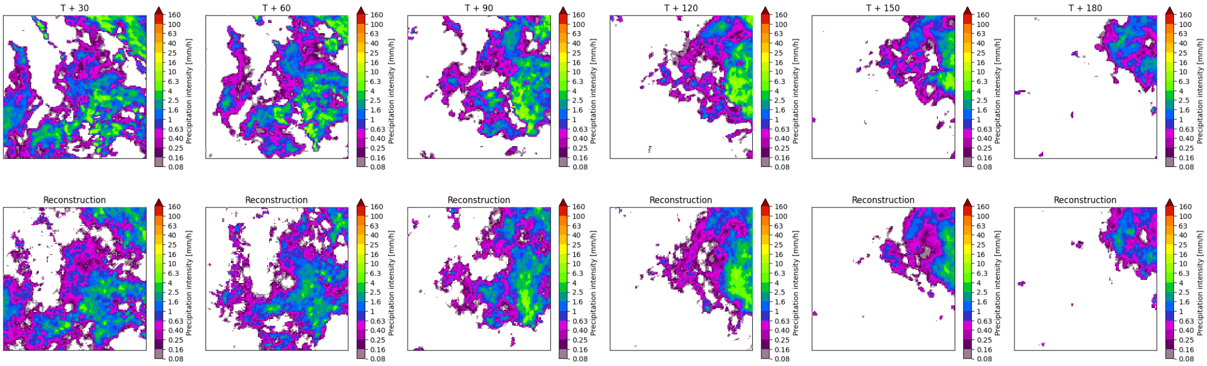


Figure A.2: Example of reconstruction of Precipitation fields by the VQ-VAE (Upper row: Original Radar Input image, Lower row: Reconstructed precipitation fields.

Appendix B

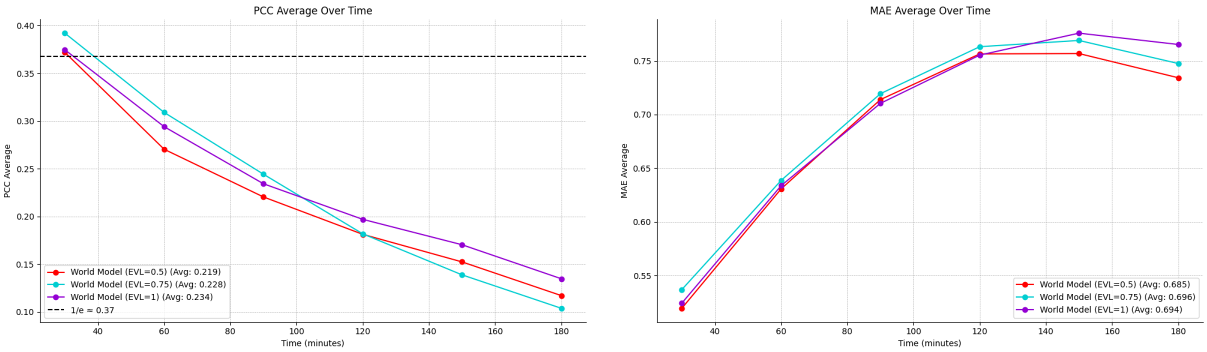
B.1 Analysis of different weights of the EVL loss function

In this section, three different weights of the EVL loss function have been implemented and their corresponding nowcasting performance on the whole Netherlands region have been compared and assessed. As mentioned in section 4.2.1, the autoregressive transformer of the **World Model (EVL)** has been incorporated with the EVL loss function using the below-mentioned loss function:

$$\mathcal{L}_{\text{Transformer-EVL}} = \mathcal{L}_{\text{Transformer}} + \lambda[\text{EVL}(u_t, v_t)]. \tag{B.1}$$

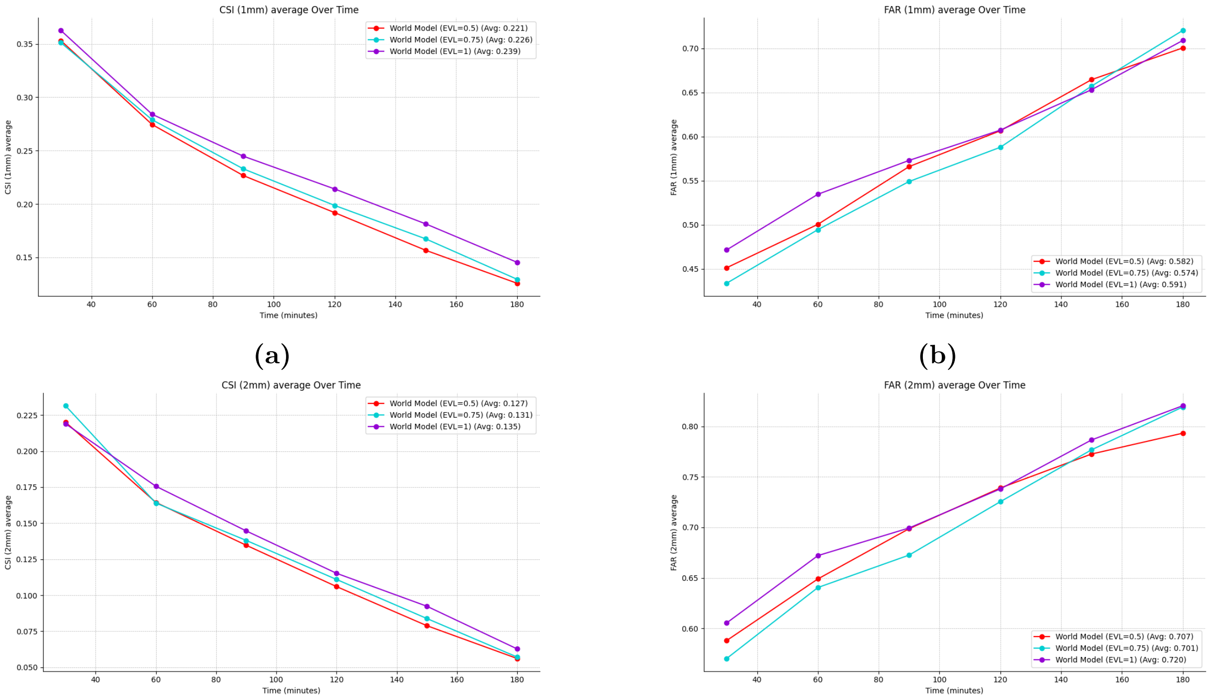
The weighting parameter λ of the EVL loss function in the above equation B.1 has been analysed using three different values - 0.5, 0.75, and 1. The graphs below show the respective model's performance with these three different weighting parameters.

Based on the below-mentioned graphs, $\lambda = 1$ has been chosen as the default setting for the **World Model (EVL)** since it shows an overall better performance when compared with the other weighting parameter configurations.



(a) **(b)**

Figure B.1: 3-hour nowcasting performance verification: continuous metrics (sub-figure (a) for PCC and sub-figure (b) for MAE). Relationship between lead time (mins) and metric scores, with 3-hour averaged scores shown in the legend.



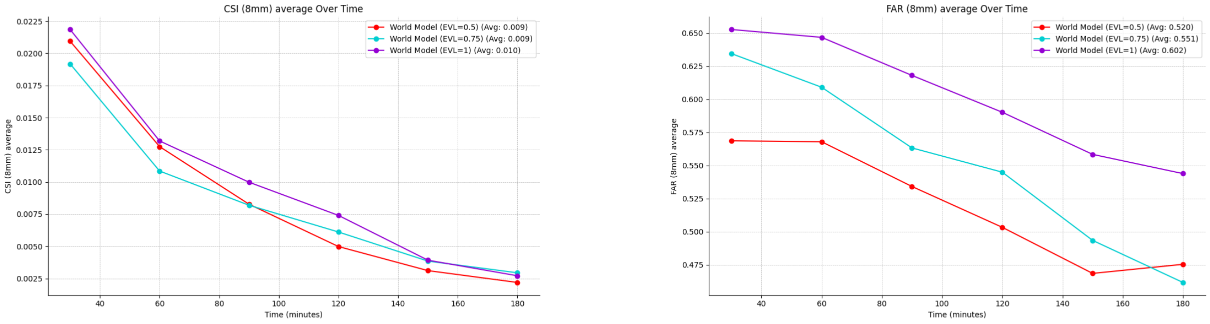
(a)

(b)

(c)

(d)

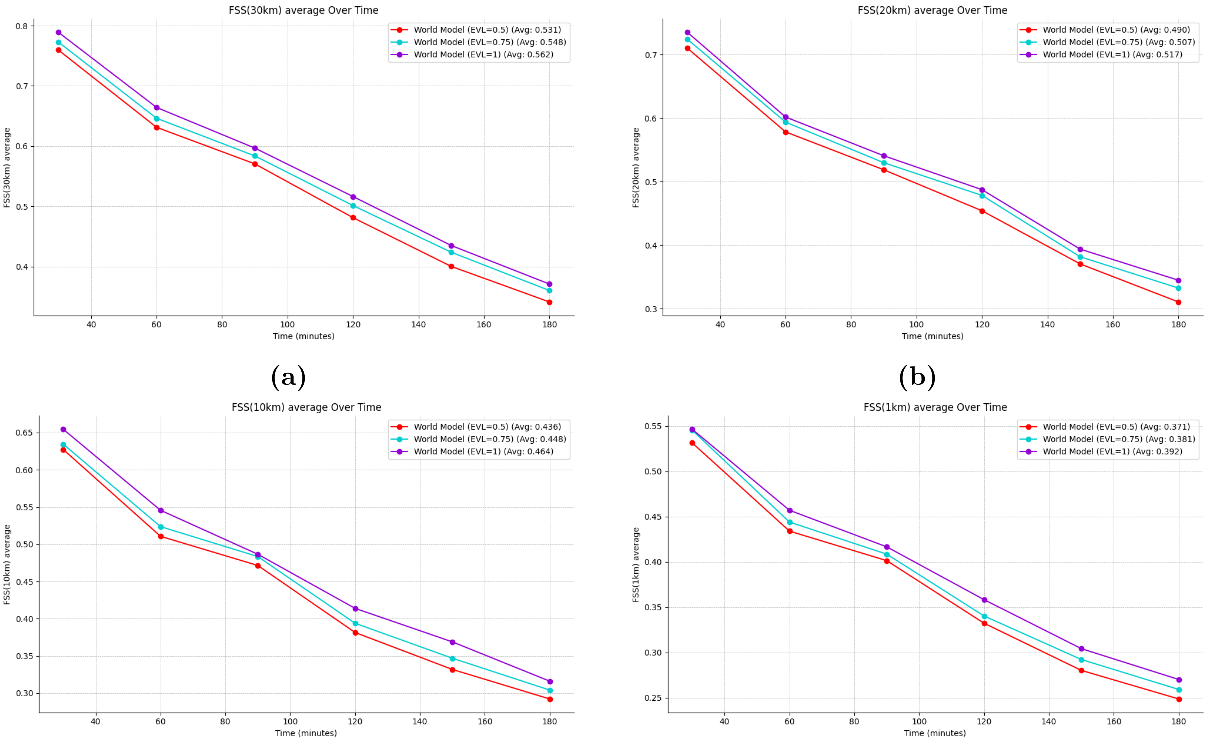
Figure B.2: 3-hour Nowcasting performance verification on the whole Netherlands region: Categorical Scores - CSI and FAR for a, b) 1mm and c, d) 2mm. Relation between lead time and metric scores, with 3-hour averaged scores shown in the legend.



(a)

(b)

Figure B.3: 3-hour Nowcasting performance verification on the whole Netherlands region: Categorical Scores - CSI and FAR for a, b) 8mm. Relation between lead time and metric scores, with 3-hour averaged scores shown in the legend.



(a) **(b)**

(c) **(d)**

Figure B.4: 3-hour Nowcasting performance verification on the whole Netherlands region: FSS Scores for a) 30km, b) 20km, c) 10km, d) 1km. Relation between lead time and metric scores, with 3-hour averaged scores shown in the legend.

Appendix C

C.1 Comparison of Training and Generation time of the different models

In this section, the training time, generation time and the number of parameters of the different models are compared for nowcasting purposes.

	Nuwa-PyTorch (VQGAN)	World Model	PySTEPS	World Model (EVL)
Number of parameters	772,832 M	402,735 M	-	520,374 M
Training time	672 h	**240 h**	-	264 h
Generation time	322.86 s	38.90 s	**9.34 s**	43.10 s

Table C.1: Comparison of training time, generation time, and the number of parameters for different models.

In the above-mentioned table, it can be observed that the generation time of PySTEPS is the fastest time followed by **World Model**. This is one of the reasons for PySTEPS being chosen as a nowcasting benchmark apart from its promising predictions. Moreover, with respect to deep learning models, it can also be observed that the generation time of **Nuwa-PyTorch (VQGAN)** is quite high compared to others. One of the main reasons for this is the application of the KV-caching (section 4.1.2) in the case of **World Model** and **World Model-EVL**. This helps in the faster generation of tokens from the autoregressive transformer thus, improving the overall generation time for the prediction frames.

Furthermore, for efficient training of the deep learning models - **World Model** and **World Model (EVL)**, 16-bit AMP as well as the conversion of the input radar maps into NumPy Arrays (for all the datasets) have also been implemented. The improvement in the training time of these two models compared to **Nuwa-PyTorch (VQGAN)** can be observed in the above table.

Appendix D

D.1 Examples of Nowcasting Predictions by different models

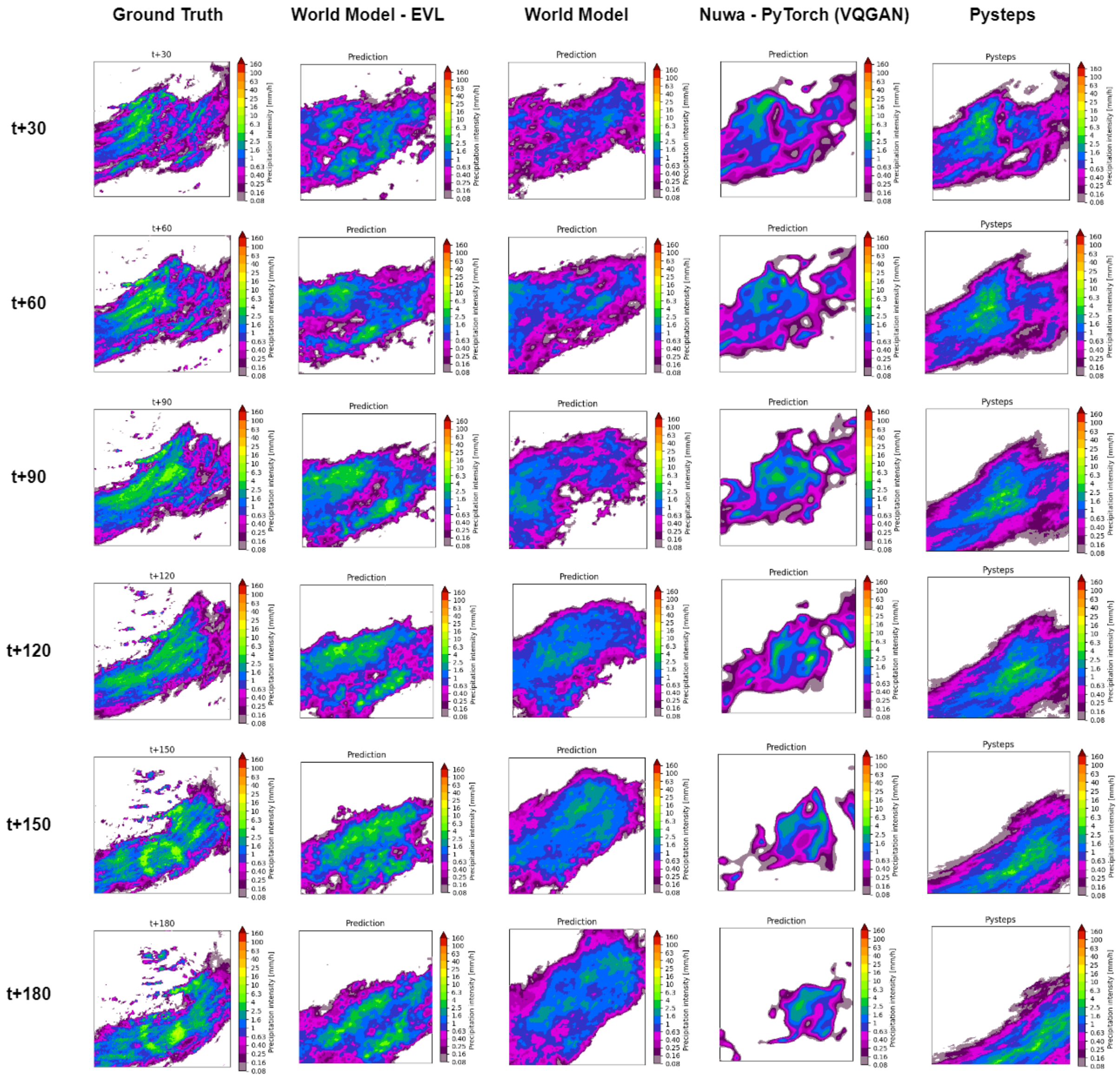


Figure D.1: Comparison of nowcasting results over the entire Netherlands region by different models (t = 05:45 2019/11/28).

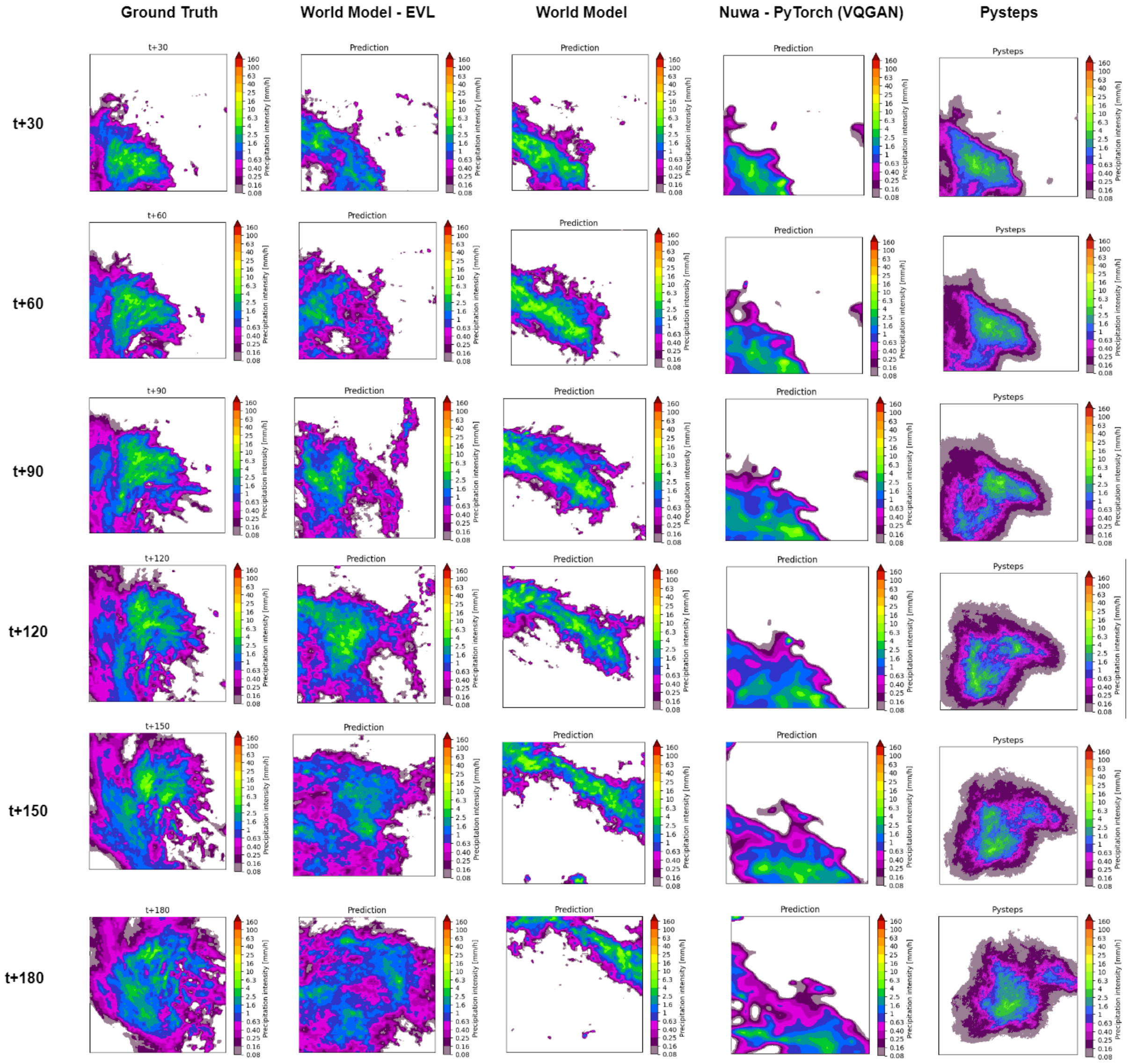


Figure D.2: Comparison of nowcasting results over the entire Netherlands region by different models (t = 03:00 2019/03/07).

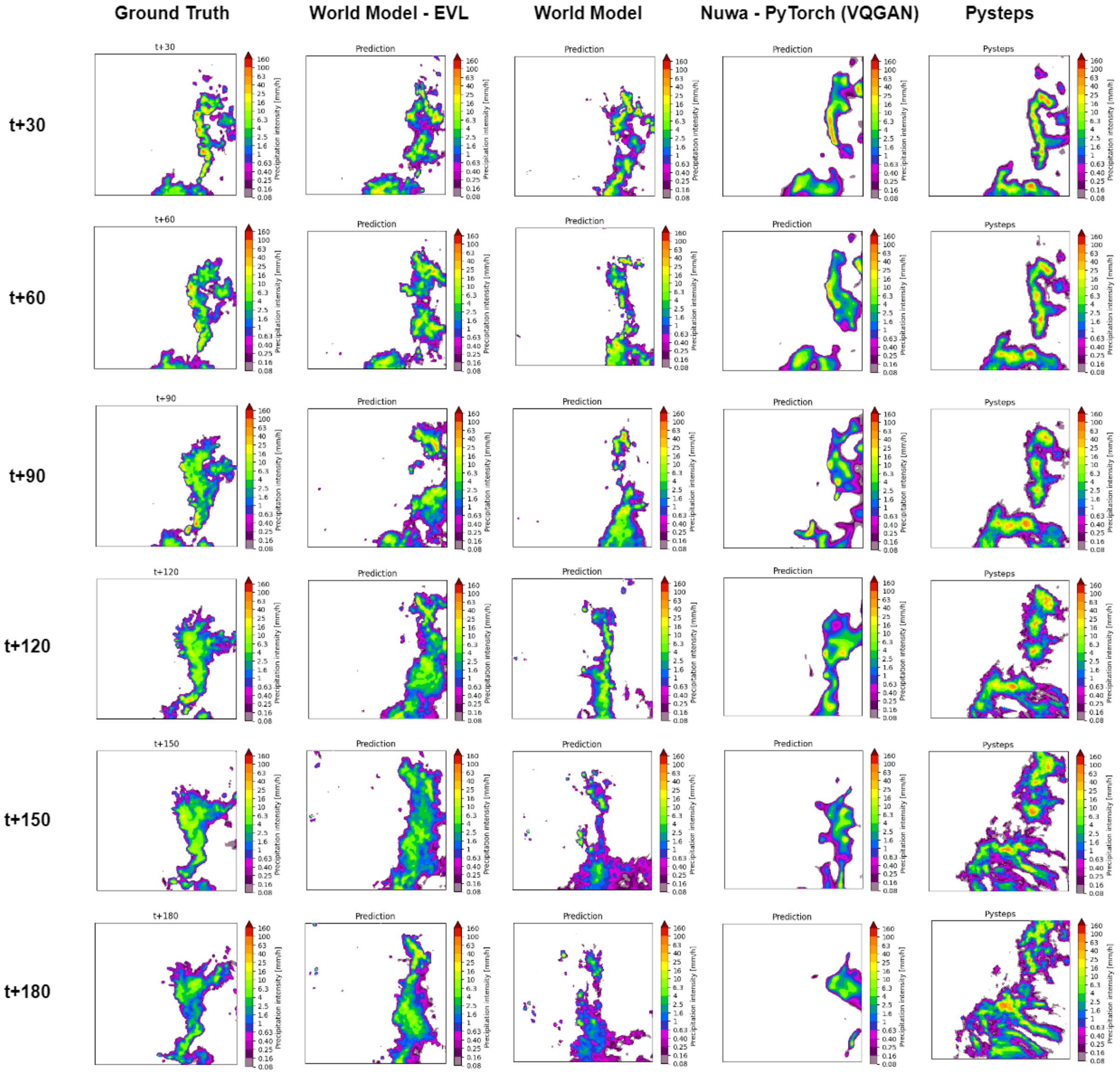


Figure D.3: Comparison of nowcasting results over the entire Netherlands region by different models (t = 16:30 2019/05/19).