ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Precipitation nowcasting with generative diffusion models

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Asperti, A., Merizzi, F., Paparella, A., Pedrazzi, G., Angelinelli, M., Colamonaco, S. (2025). Precipitation nowcasting with generative diffusion models. APPLIED INTELLIGENCE, 55(2), 1-21 [10.1007/s10489-024-06048-y].

Availability:
This version is available at: https://hdl.handle.net/11585/1000000 since: 2024-12-29

Published:

DOI: http://doi.org/10.1007/s10489-024-06048-y

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

15 December 2025

arXiv:2308.06733v2 [cs.LG] 5 Sep 2023

Precipitation nowcasting with generative diffusion models

Andrea Asperti[1*†], Fabio Merizzi[1*†], Alberto Paparella[1†], Giorgio Pedrazzi[2], Matteo Angelinelli[2] and Stefano Colamonaco[1]

[1*]Department of Informatics: Science and Engineering (DISI), University of Bologna, Mura Anteo Zamboni 7, Bologna, 40126, Italy .
[2]HPC Department Cineca, Magnanelli 6/3, Casalecchio di Reno (BO), 40033, Italy .

*Corresponding author(s). E-mail(s): andrea.asperti@unibo.it; fabio.merizzi@unibo.it;
Contributing authors: alberto.paparella2@studio.unibo.it; g.pedrazzi@cineca.it;
m.angelinelli@cineca.it; stefano.colamonaco@studio.unibo.it;
†These authors contributed equally to this work.

Abstract

In recent years traditional numerical methods for accurate weather prediction have been increasingly challenged by deep learning methods. Numerous historical datasets used for short and medium-range weather forecasts are typically organized into a regular spatial grid structure. This arrangement closely resembles images: each weather variable can be visualized as a map or, when considering the temporal axis, as a video. Several classes of generative models, comprising Generative Adversarial Networks, Variational Autoencoders, or the recent Denoising Diffusion Models have largely proved their applicability to the next-frame prediction problem, and is thus natural to test their performance on the weather prediction benchmarks. Diffusion models are particularly appealing in this context, due to the intrinsically probabilistic nature of weather forecasting: what we are really interested to model is the *probability distribution* of weather indicators, whose expected value is the most likely prediction. In our study, we focus on a specific subset of the ERA-5 dataset, which includes hourly data pertaining to Central Europe from the years 2016 to 2021. Within this context, we examine the efficacy of diffusion models in handling the task of precipitation nowcasting. Our work is conducted in comparison to the performance of well-established U-Net models, as documented in the existing literature. Our proposed approach of Generative Ensemble Diffusion (GED) utilizes a diffusion model to generate a set of possible weather scenarios which are then amalgamated into a probable prediction via the use of a post-processing network. This approach, in comparison to recent deep learning models, substantially outperformed them in terms of overall performance.

keywords: diffusion models, precipitation nowcasting, ensemble diffusion, weather forecasting, post-process

1 Introduction

The term *nowcasting* refers to the forecasting of weather indicators on a short-term meteo-scale period, typically between 2 and 6 hours. The forecast is an extrapolation in time of known weather parameters, comprising information obtained by means of remote sensing, radar echoes, and satellite data.

While numerical weather prediction models may accurately forecast the likelihood and overall

intensity of precipitation across large geographical areas and medium-term temporal intervals, the situation is more challenging when it comes to short spatial and temporal scales [1]: on less than 4 hours predictions, they frequently perform worse than persistence-based forecasts [2]. This is mostly due to the high stochastic nature of the phenomenon, in conjunction with the extended computational time typically required by numerical methods, due to the need to assimilate large amounts of data and incorporate them in the initial conditions of the models. The challenge becomes especially prominent in the case of convective precipitations, characterized by high rainfall rates originating from cells spanning a few tens of kilometers [3].

Numerous historical datasets are accessible for short and medium-range weather forecasts, and they commonly exhibit a structured spatial grid format that bears a striking resemblance to images. Each weather variable can be represented as a map, or when considering the time dimension as a video. The problem of predicting the next frame in a video sequence is a well-known image processing problem, and various categories of generative models have demonstrated their effectiveness in predicting the next frame in video sequences [4–7]; it is not surprising that many recent studies focused on the use of deep neural network (DNN) architectures for weather nowcasting [8–15]. All these models do not rely on explicit physical laws describing the dynamics of the atmosphere, adopting instead a backpropagation-based learning method to directly forecast the weather using observed data.

Most of the aforementioned models are trained to minimize the loglikelihood of the prediction, measured through a metric like mean squared error (MSE). As it is well known by other applications in image processing, loglikelihood-based optimization in the case of multimodal output typically results in averaging, introducing blurriness in the prediction. As the lead time increases, the predicted fields become weaker and more widespread, suffering from the growing uncertainty in weather predictability.

Instead of predicting the expected amount of precipitation, it is possible to address its probability distribution, which is precisely the purpose of generative modeling [16, 17]. Usually, the probability distribution is learned in an implicit way, in the form of a *generator* able to sample data in accordance with the given distribution.

For several years, the most representative class of generative models has been that of Generative Adversarial Networks (GANs) [18, 19]. In the case of GANs, the training process involves a generator, which acts as a sampler for the intended distribution, and a discriminator, responsible for evaluating the generator's output by discerning between real and generated ("fake") data. This training can be conceptualized as a zero-sum game, where the gain of one agent corresponds to the loss of the other. The generator and discriminator are alternately trained, with each adversarial component being frozen during the training of the other. Ultimately, the objective is for the generator to prevail, generating samples that the discriminator is unable to differentiate from real data. GANs typically offer better generative quality than likelihood-based models like Variational Autoencoders [20–22], possibly at the price of a reduced sampling diversity [23]. They are, in fact, prone to *mode collapse* [24], in which the generator learns to output just one or a few different examples, for instance, ignoring its noise input, and therefore generating identical outputs for a given input.

In the field of weather forecasting, GANs have been used for downscaling [25–27], precipitation estimation from remote satellite sensors [28, 29], and disaggregation [30]. The Deep Generative Models of Rainfall (DGMR) [31] is usually reputed to be the best generative nowcasting model based on GANs: it uses a conditional GAN with a regularization term to incentivize the model to produce forecasts close to the true precipitation.

The leading role traditionally held by GANs in the field of generative modeling has been recently challenged by Denoising Diffusion Models (DDM) [32]. This generative technique possesses distinctive characteristics that have been crucially exploited in many recent and well-known applications [33, 34], comprising video generation [35, 36]. These attributes include exceptional generation quality, a strong sensitivity and responsiveness to conditioning, diverse sampling capabilities, training stability, and satisfactory scalability [37, 38].

In essence, a diffusion model trains a single network to eliminate noise from images, where the level of noise to remove can be parametrically set. The network is then used to generate new samples

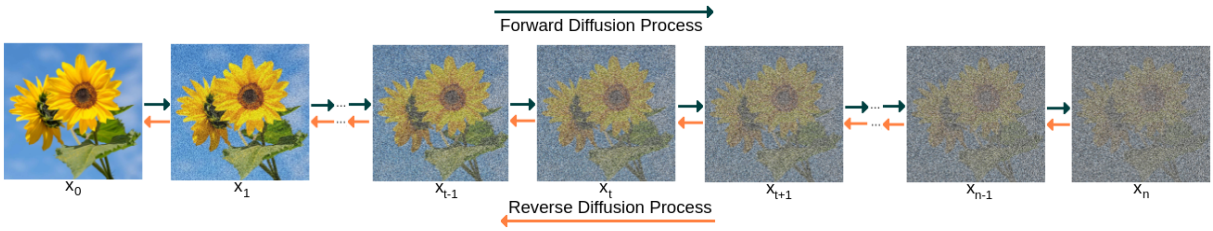


Fig. 1: Forward (from left to right) and reverse (from right to left) diffusion process.

by progressively reducing, in an iterative loop, the amount of noise in a given "noisy" image, starting from a completely random noise configuration. Traditionally referred to as reverse diffusion, this process aims to "invert" the direct diffusion process, which consists in iteratively adding noise to the source image (see Figure 1).

The idea of applying diffusion models for weather forecasting is quite natural, and several teams are independently working around this problem at the moment, on different datasets. In [39] diffusion models have been successfully applied to a downscaling problem. In [40], a diffusion model has been trained on a dataset of the MeteoSwiss operational radar network [41, 42], and tested over data obtained from the radar composite of the German Weather Service (DWD) [43] from April–September 2022. Results have been compared with a GAN-based Deep Generative Models of Rainfall (DGMR) and a statistical model, PySTEPS, showing a sensible improvement both in accuracy and diversification. A detailed comparison of this architecture with our model is given in Section 4.4.

In this article, we compare a different diffusion model with the Weather Fusion UNet (WF-UNet) model in [44], integrating precipitation and wind speed variables as input of the learning process. Data refer to six years of precipitation and wind radar images from Jan 2016 to Dec 2021 of 14 European countries, with 1-hour temporal resolution and 31 square km spatial resolution based on the traditional ERA5 dataset [45]. On the typical Mean Squared Error metric, our diffusion model outperforms WF-UNet.

The article follows the subsequent structure. It begins by addressing theoretical aspects related to diffusion models and relative conditioning. Subsequently, it introduces the dataset and experimental setting. The methodology is then presented, showcasing the novel Generative Ensemble Diffusion (GED) approach, and comparing it with related models. The article further discusses the conducted experiments and concludes by delving into the implications of the research.

2 Diffusion Models

Diffusion models are a class of probabilistic generative models that are particularly effective in modeling complex, high-dimensional data distributions and have found applications in various domains such as computer vision, natural language processing, and generative art. At the core of diffusion models lies the mathematical concept of a diffusion process, that is, a stochastic process that describes the continuous random movement of particles over time, modeling the spread or diffusion of some quantity in space or time, where the particles tend to move from regions of high concentration to regions of low concentration, resulting in a gradual blending or mixing of the quantity. In the context of machine learning, diffusion models leverage the principles of diffusion processes to model the generation of data. Instead of directly sampling data points from a fixed distribution, these models iteratively transform a simple initial distribution, typically a known distribution like a Gaussian or uniform distribution, into the desired complex data distribution. The main idea is to perform a series of diffusion steps, where each step updates the probability distribution of the data. This is achieved by adding Gaussian noise to the current data samples and iteratively refining them.

From a mathematical perspective, considering a distribution $q(x_0)$ which generates the data, generative models aim to find a parameter vector θ such that the distribution $p_\theta(x_0)$ parameterized by a neural network approximates $q(x_0)$.

Denoising Diffusion Probabilistic Models (DDPM)[32] assume the generative distribution $p_\theta(x_0)$ to have the form

$$p_\theta(x_0)) \int p_\theta(x_{0:T}) dx_{1:T} \tag{1}$$

given a time range horizon $T > 0$. where

$$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t). \tag{2}$$

Training is traditionally based on a variational lower bound of the negative loglikelihood:

$$\begin{aligned}
&-\log p_\theta(x_0) \\
&\quad \leq -\log p_\theta(x_0) + D_{\mathrm{KL}}(q(x_{1:T}|x_0) \| p_\theta(x_{1:T}|x_0)) \\
&\quad = -\log p_\theta(x_0) + \mathbb{E}_q\Big[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})/p_\theta(x_0)}\Big] \\
&\quad = -\log p_\theta(x_0) + \mathbb{E}_q\Big[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} + \log p_\theta(x_0)\Big] \\
&\quad = \mathbb{E}_q\Big[\log q(x_{1:T}|x_0) - p_\theta(x_{0:T})\Big] = \mathcal{L}(\theta)
\end{aligned} \tag{3}$$

Differently from typical latent variable models like Variational Autoencoders (VAEs) [20–22], diffusion models employ a fixed (non-trainable) inference procedure $q(x_{1:T}|x_0)$. Additionally, latent variables are characterized by relatively high dimensionality, usually identical to the dimensions of the visible space.

In the particular case of Denoising Diffusion Implicit Models (DDIMs)[46], used in this work, the authors considered a non-Markovian diffusion process

$$q_\sigma(x_{1:T}|x_0) = q_\sigma(x_T|x_0) \prod_{t=2}^{T} q_\sigma(x_{t-1}|x_t, x_0) \tag{4}$$

where $q_\sigma(x_T|x_0) = \mathcal{N}(x_T|\sqrt{\alpha_T}x_0, (1-\alpha_T)\cdot I)$, and

$$q_\sigma(x_{t-1}|x_t, x_0) = \mathcal{N}\Big(x_{t-1}\Big|\mu_{\sigma_t}(x_0, \alpha_{t-1});\ \sigma_t^2 \cdot I\Big) \tag{5}$$

with

$$\mu_{\sigma_t}(x_0, \alpha_{t-1}) = \sqrt{\alpha_{t-1}}x_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \tfrac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1-\alpha_t}}.$$

The definition of $q(x_{t-1}|x_t, x_0)$ has been cleverly chosen to respect two important aspects of the diffusion process of DDPM: the Gaussian nature of $q(x_{t-1}|x_t, x_0)$ (once conditioned on x_0) and the fact that the marginal distribution $q_\sigma(x_t|x_0) = \mathcal{N}(x_t|\sqrt{\alpha_t}x_0;\ (1-\alpha_t)\cdot I)$, recovers the same marginals as in DDPM. As a consequence of the latter property, we can express x_t as a linear combination of x_0 and a noise variable $\epsilon_t \sim \mathcal{N}(\epsilon_t|0;\ I)$:

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon_t. \tag{6}$$

Next, we need to define a trainable generative process $p_\theta(x_{0:T})$ where $p_\theta(x_{t-1}|x_t)$ leverages the structure of $q_\sigma(x_{t-1}|x_t, x_0)$. The idea is that given a noisy observation x_t, one starts making a prediction of x_0, and then use it to obtain x_{t-1} according to equation 5, that is.

In practice, we train a neural network $\epsilon_\theta^{(t)}(x_t, \alpha_t)$ to map a given x_t and a noise rate α_t to an estimate of the noise ϵ_t added to x_0 to construct x_t. Consequently, $p_\theta(x_{t-1}|x_t)$ becomes a $\delta_{f_\theta^{(t)}}$, where

$$f_\theta^{(t)}(x_t, \alpha_t) = \frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x_t, \alpha_t)}{\sqrt{\alpha_t}}. \tag{7}$$

Using $f_\theta^{(t)}(x_t, \alpha_t)$ as an approximation of x_0 at timestep t, x_{t-1} is then obtained as follows:

$$\begin{aligned}
x_{t-1} = &\sqrt{\alpha_{t-1}} \cdot f_\theta^{(t)}(x_t, \alpha_t) + \\
&\sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta(x_t, \alpha_t)
\end{aligned} \tag{8}$$

.

As for the loss function, the term in eq.3 can be further refined expressing L_θ as the sum of the following terms [47]:

$$L_\theta = L_T + L_{t-1} + \cdots + L_0 \tag{9}$$

where

$$\begin{aligned}
L_T &= D_{\mathrm{KL}}(q(x_T|x_0) \parallel p_\theta(x_T)) \\
L_t &= D_{\mathrm{KL}}(q(x_t|x_{t+1}, x_0) \| p_\theta(x_t|x_{t+1})) \\
&\qquad\qquad \text{for } 1 \leq t \leq T-1 \\
L_0 &= -\log p_\theta(x_0|x_1)
\end{aligned}$$

All previous distributions are Gaussians and their KL divergences can be calculated in closed form,

in the Rao-Blackwellized fashion. After a few manipulations, we get to the following formulation:

$$L_t = \mathbb{E}_{t\sim[1,T],x_0,\epsilon_t}\left[\gamma_t\|\epsilon_t - \epsilon_\theta(x_t,t)\|^2\right] \quad (10)$$

that can be interpreted as the weighted mean squared error between the predicted and the actual noise a time t.

The weighting parameters are frequently ignored in practice, since experimentally the training process seems to work better without them.

The pseudocode for training and sampling is given in the following Algorithms.

Algorithm 1 Training

repeat
  $x_0 \sim q(x_0)$
  $t \sim$Uniform(1,..,T)
  $\epsilon \sim \mathcal{N}(0;\ I)$
  $x_t = \sqrt{\alpha_t}x_b + \sqrt{1-\alpha_t}\epsilon$
  Backpropagate on $||\epsilon - \epsilon_\theta(x_t,\alpha_t)||^2$
until converged

Sampling is an iterative process, starting from a purely noisy image $x_T \sim \mathcal{N}(0,I)$. The denoised version of the image at timestep t is obtained using equation 8.

Algorithm 2 Sampling

$x_T \sim \mathcal{N}(0,I)$
for $t = T,...,1$ **do**
  $\epsilon = \epsilon_\theta(x_a,x_t,\alpha_t)$
  $\tilde{x}_0 = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\epsilon)$
  $x_{t-1} = \sqrt{\alpha_{t-1}}\tilde{x}_0 + \sqrt{1-\alpha_{t-1}}\epsilon$
end for

2.1 Conditioning

Generation often requires a way to control how samples are created to influence the final output. This process is commonly referred to as *conditioned* or *guided* diffusion. Numerous approaches have been devised to integrate image and/or text embeddings into the diffusion process, allowing for guided generation. In the mathematical context, "guidance" entails conditioning a prior data distribution, represented by $p(x)$, with a particular constraint, like a class label or an image/text embedding. This conditioning leads to the formation of a conditional distribution, denoted as $p(x|y)$.

To convert a diffusion model p_θ into a conditional diffusion model, we can introduce conditioning information y at each diffusion step as follows:

$$p_\theta(x_{0:T}|y) = p_\theta(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t,y) \quad (11)$$

There are typically two approaches to learning this distribution, one based on an auxiliary classifier (similar, in spirit, to AC-GANs [48]), and a second one that is classifier-free.

The idea behind classifier guidance is the following. Our aim is to learn the gradient of the logarithm of the conditional density $p_\theta(x_t|y)$. By applying Bayes' rule, we can express it as:

$$\nabla_{x_t}\log p_\theta(x_t|y) = \nabla_{x_t}\log\left(\frac{p_\theta(y|x_t)\cdot p_\theta(x_t)}{p_\theta(y)}\right) \quad (12)$$

Since the gradient operator only applies to x_t, we can eliminate the term $p_\theta(y)$; after simplification we get:

$$\begin{aligned}\nabla_{x_t}\log p_\theta(x_t|y) =& \nabla_{x_t}\log p_\theta(x_t) \\ &+ s\cdot\nabla_{x_t}\log p_\theta(y|x_t)\end{aligned} \quad (13)$$

Here s is a scalar term used to modulate the strength of the guidance term.

As described in [37], we can use a classifier $f_\phi(y|x_t,t))$ to guide the diffusion during generation. This technique involves training a classifier $f_\phi(y|x_t,t)$ on a noisy image x_t to predict its class y. The gradient $\nabla_x \log f_\phi(y|x_t)$ can then be utilized to guide the diffusion sampling process towards the conditioning information y by modifying the noise prediction. We shall not use this technique, particularly suited for discrete labels, so further details are omitted.

The theory of condition diffusion without relying on an independent classifier has been investigated in [49]. The approach consists in training a conditional diffusion model $\epsilon_\theta(x_t,t,y)$ along with an unconditional model $\epsilon_\theta(x_t,t,0)$. Typically, the

same neural network can be used for both models: during training, the class y is randomly set to 0, exposing the model to both conditional and unconditional setups. The estimated noise $\hat{\epsilon}_\theta(x_t|t, y)$ at timestep t is then a suitably weighted combination of the conditional and unconditional predictions:

$$\hat{\epsilon}_\theta(x_t, t, y) = \epsilon_\theta(x_t, t, y) + s \cdot \epsilon_\theta(x_t, t) \quad (14)$$

3 Dataset description and preprocessing

To assess the performance of our model we recreate the same samples as proposed in [44, 50, 51], focusing on the task of precipitation nowcasting. We selected for our research a subset of the ERA-5 dataset [45], which is a state-of-the-art global atmospheric reanalysis produced by the European Centre for Medium-Range Weather Forecasts (ECMWF). ERA-5 reports a comprehensive numerical representation of the Earth's recent climate history, spanning several decades and covering the entire globe at a high spatial resolution of approximately 31 kilometers. It provides hourly estimates of a multitude of atmospheric, land, and oceanic climate variables, such as temperature, precipitation, humidity, wind speed and direction, and sea surface temperature, among others. The ERA-5 dataset is the product of an advanced and uniform data assimilation system, melding millions of disparate observations with intricate Earth system modeling. This integration yields a coherent and consistent dataset, highly regarded and extensively used across various fields. Its applications span weather forecasting, climate studies, hydrological research, energy production prediction, and numerous other scientific domains, as well as policy-related endeavors.

With our task being precipitation nowcasting, our main target feature is the Total Precipitation variable, described as the accumulated liquid and frozen water that falls to the Earth's surface, comprising of rain and snow. Our selected region of interest is defined by a geographical rectangle, with its latitudinal boundaries extending from latitude -12° to latitude 12 ° and its longitudinal boundaries extending from longitude 36° to longitude 60°. Images from this region cover much of the western part of Europe, partially covering 14 countries. Our collected time span covers a six-year period, from 2016 to 2021, with hourly measurements. The collected data have a dimension of 96×96 values with each representing the depth fallen water would have if it were spread evenly over the grid box of 31 square kilometers. The units of this parameter are depth in meters of water equivalent.

The dataset is normalized by dividing the values of both the training and testing set by the highest occurring value in the training set, we then split the dataset into a training set (years 2016-2020) and a testing set (year 2021).

Rain is a sparse parameter, and it is often non-present in the area of analysis. This produces a dataset with an extensive amount of noninformative data which can bias the model towards a zero prediction [51]. Therefore, we defined the generator for our data with an additional parameter, so that it returns only sequences with at least a percentage amount of rain in the pixels, therefore simulating the EU-50 and EU-20 datasets specified in [44], whose images have at least 50% and 20% of rain in the pixels respectively. This selection operation is performed by computing the number of non-zero pixels on a wider region of size 105x173, the image is then cropped to the final dimension of 96x96.

3.1 Additional features

In precipitation nowcasting it is often necessary for the predictive models to consider a range of meteorological features beyond the presence or absence of rain. Variables such as temperature, pressure, humidity, wind direction, and wind speed can all significantly influence precipitation patterns. These factors, among others, interact in complex ways to shape the dynamics of the atmospheric system. Another topic of concern is the level of awareness the model has of the underlying physical structure, and elements such as time embeddings, land/sea mask, and elevation information may be valuable for enabling the model to make a more informed decision. In particular, in our work we experimented with wind speed, obtained from the two different northerly and easterly wind components, the land-sea mask, a geopotential map, and a sinusoidal time embedding, as reported in Table 1. All obtained from the

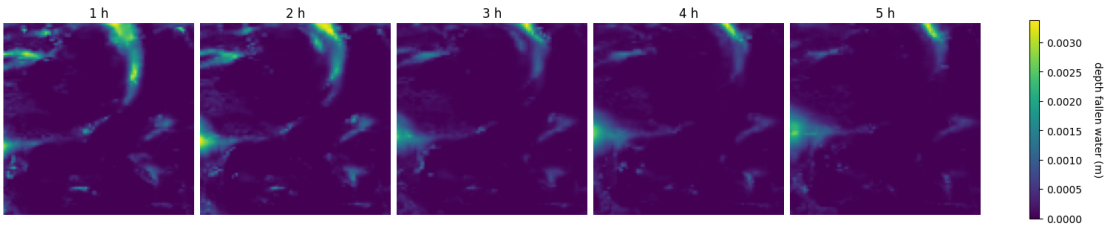


Fig. 2: Example of precipitation data from the ERA-5 dataset

ERA-5 dataset and normalized between 0 and 1 before training:

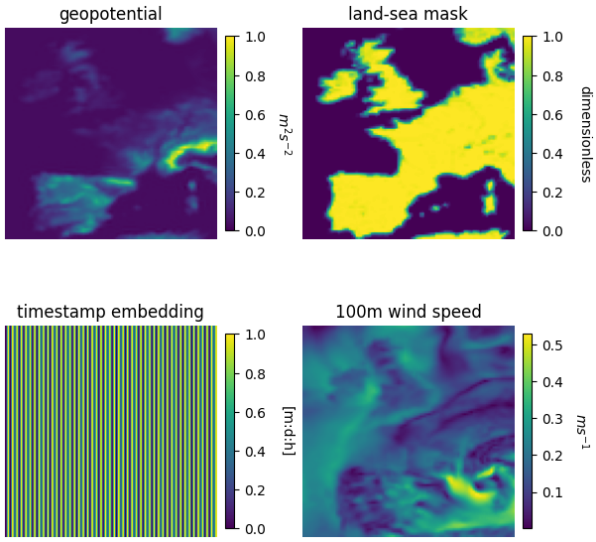


Fig. 3: Visual example of the additional features

4 Methodology

In this section, we introduce our proposed model, Generative Diffusion Ensemble (GDE). Our model is based on the Denoising Diffusion Implicit Model (DDIM), which is trained on a sequence of rain data augmented with other meteorological features. The model generates multiple outputs that are then synthesized into a final prediction through a U-Net architecture. This combination provides a comprehensive precipitation forecast, leveraging both the capability of DDIM to model the probability distribution of the weather data and the feature extraction strength of the U-Net.

We start by introducing the distinct components that form our GDE model, offering insight into their specific roles within the broader system. The discussion ultimately converges on the overall structure of the model.

4.1 The DDIM architecture

Diffusion models essentially operate as iterative denoising algorithms. Their main trainable component is the denoising network, denoted as $\epsilon_\theta(x_t, \alpha_t)$. This network receives as input the noisy images, x_t, and a corresponding noise variance, α_t, with the objective of estimating the amount of noise infiltrating the image. The training of this underlying denoising network is done conventionally. An initial sample, x_0, is extracted from the dataset and subjected to a predefined amount of random noise. The network is then tasked with estimating the noise within these corrupted images.

4.1.1 denoising

Our model of choice as the denoising network is a U-net. The U-net is one of the most common architectures for denoising [52–55] and it is often implemented in diffusion models [56]. Originally introduced for semantic segmentation [57], the U-Net architecture has gained widespread popularity and found applications in diverse image manipulation tasks. The network comprises a downsampling sequence of layers, followed by an upsampling sequence while incorporating skip connections between layers of the same size. Typically, the U-Net's configuration is determined by defining the number of downsampling blocks and the number of channels for each block. The upsampling structure follows a symmetric pattern and the spatial dimension is dependent on the image resolution, which in our case is 96x96. Consequently, a U-Net's entire structure can be encoded concisely in a single list, such as [32,

Name	Unit	Description
100m wind speed	ms^{-1}	Wind speed of air at a height of 100 meters above the surface of the Earth, given easterly and northerly components u and v the speed is obtained by $\sqrt{(u^2+v^2)}$
Timestamp	[m,d,h]	timestamp including month, day, and hour of the start of the given sequence, tile encoded into a 96x96 array.
Land-sea mask	dimensionless	Proportion of land, as opposed to ocean or inland waters in a grid box
Geopotential	m^2s^{-2}	Gravitational potential energy of a unit mass, at a particular location at the surface of the Earth, relative to mean sea level.

Table 1: Additional features units and details

64, 96, 128]. This list represents both the number of downsampling blocks (in this case, 4) and the corresponding number of channels, which usually increase as the spatial dimension decreases. For our experiments we selected a U-net size of [64, 128, 256, 384] which proved to be the most effective experimentally. To improve the sensibility of the U-net to the noise variance, α_t is taken as input, which is then embedded using an ad-hoc sinusoidal transformation by splitting the value in a set of frequencies, in a way similar to positional encodings in Transformers [58]. The embedded noise variance is then vectorized and concatenated to the noisy images along the channel axes before being passed to the U-Net. This helps the network to be highly sensitive to the noise level, which is crucial for good performance. We implement sinusoidal embeddings using a Lambda layer.

4.1.2 conditioning

Conditioning of the model is necessary to guide the diffusion towards a forecast defined by the known previous weather conditions. Our conditioning is applied in a classifier-free manner, by concatenating the conditioning frames to the noisy images alongside the channel axis.

Practically, the model $\epsilon_\theta(x_t, t, y)$ takes as input the noisy images $x_t = \{r_1, r_2, r_3\}$ where r_h represents the future prediction of the rain precipitation h hours ahead. The conditioning information $y = \{r_{-8..0}, u_{-1,0}, v_{-1,0}, lsm, geopot\}$ contains the previous 8 hours of precipitation information $r_{-11..0}$, the previous two hours for both wind components $u_{-1,0}, v_{-1,0}$ and two static maps representing the land-sea mask lsm and the geopotential $geopot$.

Our implementation directly provides the U-net with the conditioning information, specifically, each temporal slice in the input data is treated analogously to a color channel in an RGB image. By applying 2D convolutions across these temporal slices independently our model is able to extract sufficient frame-level temporal features to effectively produce a sequence coherent with the past frames used as conditioning.

For example, when training with a batch size of 16, our input data to the denoising network would have the shape of [16,96,96,17], with the last dimension containing both the conditioning information and noisy images. Our output on the other hand would comprise only the denoised 3 frames, therefore would have the shape of [16,96,96,3].

Interestingly, our training process was proven successful in consistently achieving temporal conditioning training exclusively with conditioned instances, whereas examples in the literature required alternating between conditioned and non-conditioned training instances, followed by a weight mixing stage [59].

4.2 The proposed Generative Diffusion Ensemble (GED)

Our proposed Generative Diffusion Ensemble (GED) approach leverages the power of diffusion to integrate the inherent probability distribution

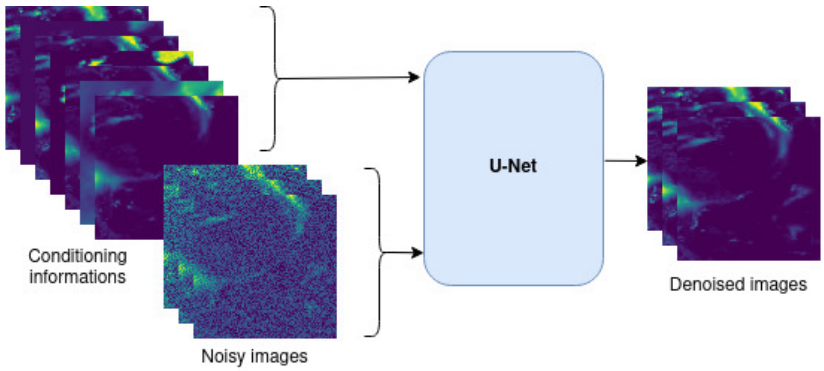


Fig. 4: Conditioning is implemented by stacking additional information alongside the channel axis in the denoising network

of meteorological patterns, which is then used to synthesize a probable precipitation prediction. Image samples generated through the diffusion process, in line with the generative traits of the model, yield a highly diverse set of outputs despite sharing identical conditioning information. Based on the assumption that the diffusion model captures the stochastic essence of weather dynamics, a prediction can be derived from an ensemble of possible outcomes.

The methodology we propose aligns fundamentally with the ensemble post-processing techniques conventionally utilized in Numerical Weather Prediction (NWP) models. Ensemble post-processing in weather forecasting refers to the statistical refinement of a set of multiple weather forecasts produced from slightly different initial conditions [60]. These multiple simulations provide a range of possible weather outcomes and function as an estimate of forecast uncertainty. The methods employed in ensemble post-processing exhibit considerable diversity and range from statistical techniques such as linear regression and distributional regression [61, 62], to more sophisticated machine learning algorithms like QRF [63] or EMOS-GB [64]. In recent years most of the research interest focused on Neural methods for ensemble post-processing, with notable successful examples in literature. [65, 66].

In our work, we propose two different approaches to executing the ensemble prediction, one based on a simple statistical method and the second utilizing a neural model. The first approach involves executing multiple diffusion generations in an iterative manner (parallelizable across the batch dimension with a significant speedup [67]) and subsequently calculating the mean of these generated images. This method effectively condenses the probability distribution of the images into an average outcome, thereby yielding a more accurate forecast. Our experimental results have demonstrated that this strategy of computing the mean of multiple generations produces superior results when compared to utilizing a single diffusion generation. Thus, this approach effectively leverages the multitude of potential outcomes to generate a more accurate and robust prediction.

In our second approach, instead of simply computing a mean, we delegate the synthesis of the prediction to a more sophisticated module, similar to what is frequently done in the literature [68, 69]. In our case, we use a U-Net architecture to amalgamate the generated outcomes into a more probable prediction. This strategy not only allows for a more informed decision-making process in integrating the outputs but also provides the opportunity to add a post-processing layer specifically trained on our target image, in contrast to the diffusion model which is trained on the noise difference of the single diffusion steps. Our experimental findings have validated the efficacy of this methodology, demonstrating it to consistently produce superior results to both simple diffusion and diffusion ensemble with the mean method.

4.3 Training and Evaluation

Our diffusion model underwent training with a batch size of 2 over 40 epochs, using the AdamW

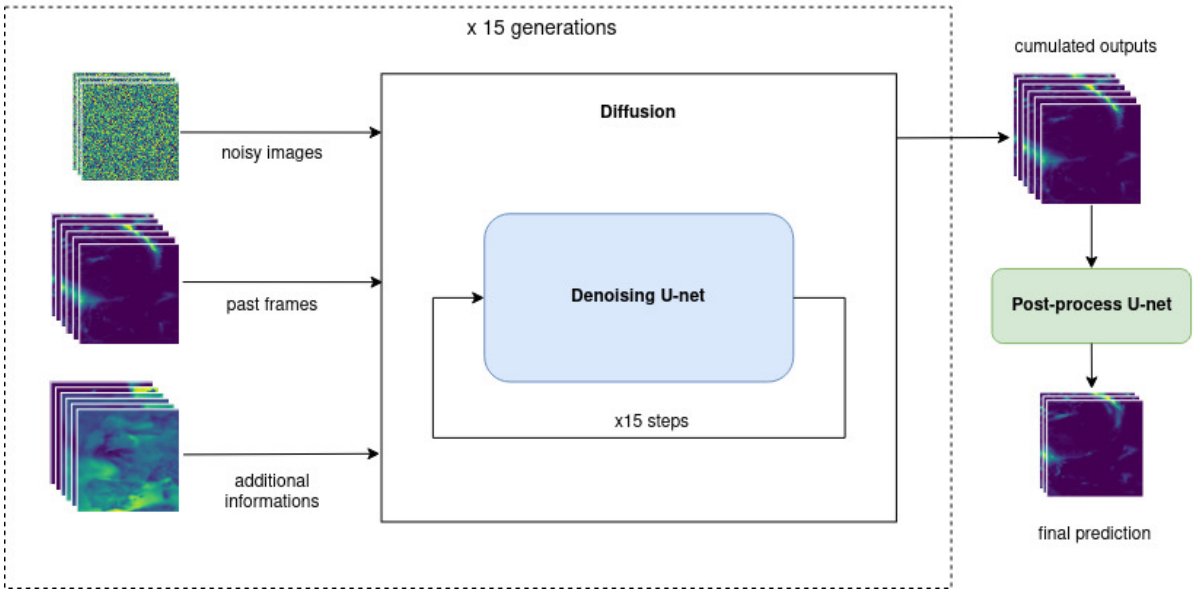


Fig. 5: Generative Diffusion Ensemble (GED) prediction structure, showing the multiple denoising cycles and the final post-processing step.

optimization algorithm [70] with a learning rate of 1e-04 and weight decay of 1e-05. Interestingly, we observed that augmenting the batch size hampered the training process. Furthermore, a fine-tuning phase of 10 epochs using a learning rate of 1e-05 and weight decay of 1e-06 yielded marginal improvements in the overall results. Training was implemented via the use of a generator, which produced a random batch of sequences spanning the training years from 2016 to 2020. Mirroring our reference model, we omitted sequences composed of over 50% non-rain values from the training process. Consistent with standard diffusion model implementations, our loss function was the Mean Absolute Error (MAE) applied to the difference in noise. Training experiments based on image loss, as opposed to noise, led to inferior results.

Our diffusion model underwent evaluation based on data from the test year of 2021. The number of diffusion steps was fixed at 15, with performance assessed using the Mean Squared Error (MSE) metric, defined in Equation 15. Here, n represents the total number of samples, y_i denotes the ground truth value, and $\hat{y}_i$ signifies the predicted value. Please note that all metrics were computed on data post-denormalization. All training was conducted on an Nvidia RTX 4000 graphic card using the TensorFlow/Keras framework.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{15}$$

The Generative Diffusion Ensemble (GDE) incorporates a post-processing U-net that shares the spatial dimensions of the denoising U-net, excluding the embedded variance. This network accepts fifteen distinct generative outputs from the diffusion model as input, each consisting of three future predictions for the subsequent three hours. The U-net then yields an output comprising three images, each predicting the rainfall for one of the upcoming three hours.

Training of this model parallels the process used for our diffusion model, employing AdamW as the optimizer with a learning rate of 1e-4 and a weight decay of 1e-5. The loss is computed using the Mean Squared Error (MSE) between the predicted images and their corresponding ground truth. For training, data is dynamically generated by the diffusion model using random sequences from the training years. Likewise, evaluation is performed on sequences from the test year of 2021.

4.4 Differences with related models

To the best of our knowledge, the only other work currently addressing precipitation nowcasting using diffusion models is [71]. Comparing the two architectures is quite challenging due to the significantly different spatiotemporal scales of the data involved. In [71], they consider time steps of 5 minutes, utilizing 4 time steps (20 minutes) of precipitation as input, and predict precipitation up to 20 time steps (100 minutes) into the future. Moreover, the geographical scale differs greatly as well, with radar signals collected at a 1 km resolution in a rectangular area spanning 710 km in the east–west direction and 640 km in the north–south direction, covering all of Switzerland and some surrounding regions.

In the cited work, the network is trained on 256x256 pixel images, corresponding to a geographical area of 256 Km^2. To manage computational costs, they adopt the notion of Latent Diffusion Model (LDM) popularized by Stable Diffusion [72], where the diffusion process runs in a latent variable space mapped to the physical pixel space through an autoencoder.

The diffusion model used in their work is quite different from ours, starting with the number of denoising iterations, which is 50 compared to our 15. Additionally, their Denoiser makes use of a forecaster stack based on Adaptive Fourier Neural Operators (AFNOs) [73][74] to condition the model. They incorporate temporal cross-attention to map between the input and output time coordinates and a different denoiser stack that is also based on AFNOs, simulating cross-attention. The actual relevance of these modules is not documented, since no ablation was performed. We tested some of these solutions without noticing sensible improvements.

Other related works in precipitation nowcasting based on CNNs often utilize 3D convolutions or other conditioning methods such as RNNs or LSTMs to explicitly model temporal connections [44, 50, 51, 75]. In recent years multiple publications have shown promising results in treating timesteps as multiple channels in the network, in this way achieving temporal prediction with only 2D convoluted layers [76, 77]. In our proposed diffusion model, we handle temporal data using 2D convolutions in a similar manner to how color channels are dealt with in image processing. Specifically, each temporal slice in the input data is treated analogously to a color channel in an RGB image. By applying 2D convolutions across these temporal slices independently, akin to processing different color channels, our model was able to match the performance of competing 3D CNNs models.

5 Experiments and Results

In this section, we present our experimental setup and the conducted experiments.

All experiments were performed using models implemented in the TensorFlow/Keras framework. The training dataset consisted of precipitation data and additional features for the specified region, ranging from 2016 to 2021, while the test set exclusively utilized data collected in 2021. To compute the results, we performed an exhaustive analysis of all sequences within the given year for both the EU-20 and EU-50 datasets.

All of our models compute the three different predictions simultaneously. It may be argued that with a unique model instance for each prediction, the overall performance may slightly improve, but this would happen at the expense of the training and inference time, with the latter being especially relevant for the operational application of precipitation forecasting.

Our primary goal is to minimize the Mean Squared Error (MSE) within the initial three hours of the prediction model. Nonetheless, the determination of the optimal input data and model configuration remains a topic of ongoing debate. To address this, we conducted a series of initial tests aimed at identifying the most advantageous set of input features. In Table 4 we report a comparison of scores obtained with different sets of additional features using the Single Diffusion model.

For what concerns model comparison, Table 2 and 3 present a comparative analysis between a standard Core U-Net model, BroadU-Net [50], WF-UNet (which incorporates additional features as proposed by [44]), our diffusion model with a singular generative output (Single Diffusion), and two distinct implementations of the Generative Ensemble Diffusion (GED). The GED models generate a final prediction by integrating 15 different generations of the three predicted frames.

MSE values and additional metrics for EU20 dataset				
Model	**MSE**	**Accuracy**	**Precision**	**Recall**
1 hour ahead				
Core U-net	2.97e-04	0.863	0.698	0.837
Broad U-net	3.05e-04	0.861	0.706	0.803
WF-UNet	2.67e-04	0.933	0.790	0.847
Single Diffusion	2.86e-04	0.911	0.754	0.888
GED (mean)	2.25e-04	0.930	0.786	0.901
GED (postprocess)	2.03e-04	0.923	0.798	0.909
2 hour ahead				
Core U-net	5.02e-04	0.813	0.609	0.796
Broad U-net	5.05e-04	0.819	0.638	0.712
WF-UNet	4.87e-04	0.895	0.664	0.807
Single Diffusion	4.69e-04	0.886	0.705	0.831
GED (mean)	3.93e-04	0.900	0.731	0.848
GED (postprocess)	3.53e-04	0.898	0.742	0.849
3 hour ahead				
Core U-net	6.71e-04	0.800	0.612	0.657
Broad U-net	6.55e-04	0.806	0.637	0.609
WF-UNet	6.34e-04	0.877	0.626	0.736
Single Diffusion	6.10e-04	0.853	0.638	0.758
GED (mean)	5.20e-04	0.880	0.689	0.801
GED (postprocess)	4.70e-04	0.891	0.701	0.796

Table 2: Results comparison on the EU20 Dataset

In the first implementation, the prediction is calculated by averaging all the values (mean), while the second version employs a post-processing U-Net for this task (post-process). The primary metric utilized for these results is Mean Squared Error (MSE), supplemented with the additional metrics of Accuracy, Precision, and Recall. Our findings suggest that, although a single diffusion prediction is outperformed by the U-Net models, both implementations of GED significantly surpass the performance of any U-Net-based approach. Among these, the GED version with post-processing demonstrated the most superior overall performance.

The diffusion results for the test set of 2021 reveal a significant dissimilarity in performance throughout the year. Figure 6a distinctly illustrates this dissimilarity for all three forecasted timeframes. Moving to Figure 6b, it becomes evident that this dissimilarity persists whether the prediction is generated using Single Diffusion, GDE (mean), or GDE (post-process). This consistency may be due to the intricacies inherent in precipitation forecasting, which can be especially demanding during specific periods of the year. Precipitation patterns are notably prone to robust seasonal fluctuations [78]. For example, shifts between seasons can trigger abrupt atmospheric changes, complicating the precise prediction of precipitation types and quantities [79]. Convective precipitation, which is more common during warm and humid conditions in the warmer months of the year, is closely linked with swiftly evolving weather systems like thunderstorms. It remains a formidable challenge for accurate forecasting, even in traditional operational meteorology [80, 81], as these systems manifest rapidly and

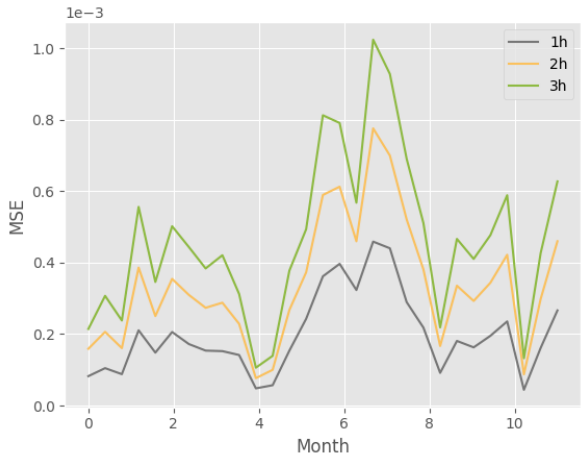
MSE values and additional metrics for EU50 dataset				
Model	**MSE**	**Accuracy**	**Precision**	**Recall**
1 hour ahead				
Core U-net	3.18e-04	0.862	0.698	0.833
Broad U-net	3.24e-04	0.860	0.705	0.795
WF-UNet	2.50e-04	0.921	0.803	0.849
Single Diffusion	2.59e-04	0.915	0.767	0.882
GED (mean)	2.02e-04	0.924	0.782	0.885
GED (postprocess)	1.99e-04	0.913	0.803	0.907
2 hour ahead				
Core U-net	5.02e-04	0.813	0.609	0.796
Broad U-net	5.05e-04	0.819	0.638	0.712
WF-UNet	4.62e-04	0.877	0.684	0.813
Single Diffusion	4.51e-04	0.875	0.699	0.844
GED (mean)	3.59e-04	0.882	0.711	0.862
GED (postprocess)	3.40e-04	0.878	0.724	0.860
3 hour ahead				
Core U-net	6.71e-04	0.800	0.612	0.657
Broad U-net	6.55e-04	0.806	0.637	0.609
WF-UNet	6.31e-04	0.855	0.647	0.743
Single Diffusion	6.03e-04	0.848	0.672	0.801
GED (mean)	4.92e-04	0.856	0.701	0.828
GED (postprocess)	4.65e-04	0.861	0.706	0.821

Table 3: Results comparison on the EU50 Dataset

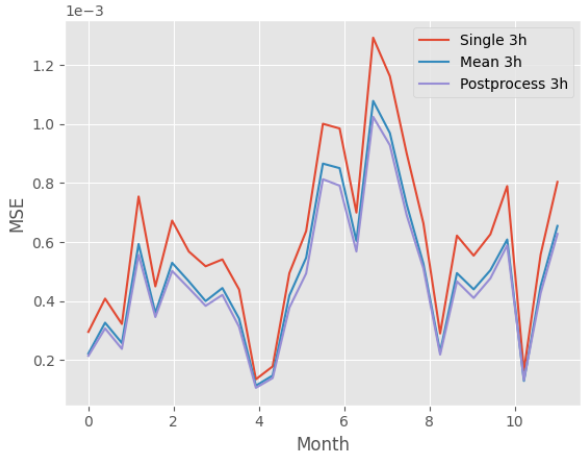
Single diffusion with different inputs on EU-50			
Inputs	**MSE 1h**	**MSE 2h**	**MSE 3h**
8 rain	2.62e-04	4.60e-04	6.21e-04
8 rain + lsm + geopot	2.60e-04	4.61e-04	6.23e-04
8 rain + lsm + geopot + time	2.60e-04	4.56e-04	6.16e-04
8 rain + lsm + geopot + time + 2 wind speed	2.59e-04	4.51e-04	6.03e-04

Table 4: Results comparison on the EU20 Dataset

are subject to an array of intricate and dynamic atmospheric processes. This led to the emergence in the literature of proposals to tackle this specific task, namely severe weather especially in regions more subject to these phenomena Additionally, the transit of weather fronts, encompassing cold and warm fronts, is more common during transitional seasons, such as spring and fall, when the differences in temperature and air masses are more pronounced [82]. This phenomenon exerts swift alterations in precipitation distribution. The interplay between distinct air masses within these



(a)



(b)

Fig. 6: Single diffusion results for the year 2021 on EU50, showing high dissimilarity in score depending on the month of the year. In (a) we can see that the dissimilarity is present for each of the 3 predicted hours, in (b) note that the dissimilarity is not affected by computing the outcome with Single Diffusion, GDE (mean), or GDE (post-process).

fronts introduces complexity, making the precise timing and location of precipitation a challenging problem.

6 Conclusions

In this study, we tackled the challenging task of precipitation nowcasting using diffusion models. Through experimentation on well-established tasks documented in the literature, our proposed Generative Ensemble Diffusion (GED) approach achieved a greater overall performance with respect to competing U-Net based models.

Our primary objective was to explore the hypothesis that a diffusion model could effectively capture the intricate chaotic behaviors of weather patterns by modeling its probability distribution. We worked on a subset of the ERA-5 dataset utilizing hourly data from a region of central Europe, training on the years 2016-2020 and testing on 2021. We trained our models on pre-processed versions of the dataset matching known works in literature [44].

Our experimental findings reveal that the incorporation of additional weather features enhances the prediction quality. Specifically, the inclusion of wind speed, land-sea mask, timestamp, and geopotential data contributed to improving the overall quality of our predictions.

Initially, our diffusion model's single generative predictions demonstrated a lower performance compared to a compatible prediction conducted using a U-net. However, a remarkable breakthrough emerged when we computed multiple generative predictions in parallel. By skillfully amalgamating these diverse outcomes through a post-processing step, we achieved a substantial improvement in the prediction quality, surpassing the performance of well-established U-net models.

The probabilistic nature of the model, combined with its use of ensemble forecasting, makes it well-suited for forecasting rare events which occur with low probability but have a significant impact on the population and the economy [83].

Overall, our work contributes to the advancement of precipitation nowcasting methodologies and offers a promising perspective on leveraging diffusion models to gain a deeper understanding of weather phenomena. The combination of GED with post-processing demonstrates the potential to enhance weather precipitation nowcasting, providing valuable information for various applications and decision-making processes.

A distinctive aspect of our work is that it has been conducted with very limited computing resources. The majority of computations were performed on a single workstation, which was equipped with a GPU Quadro RTXA4000 with 16GB of VRAM and 32GB of RAM.

Nevertheless, our research is part of an ongoing collaboration with the High Performance Computing Department of Cineca. As a next step, we intend to evaluate the model using the state-of-the-art Leonardo system, which offers significantly greater computational power. Specifically, our plan involves testing diffusion models on more intricate weather benchmarks. These benchmarks encompass predictions at varying spatial and temporal scales, with a particular focus on medium and long-term temporal ranges.

In addition, in the context of the European Cordis Project "Optimal High Resolution Earth System Models for Exploring Future Climate Changes", we plan to apply our methodology to downscaling of meteorological indicators.

Acknowledgements

This research was partially funded and supported by the following Projects:

- European Cordis Project "Optimal High Resolution Earth System Models for Exploring Future Climate Changes" (OptimESM), Grant agreement ID: 101081193
- Future AI Research (FAIR) project of the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.3 funded from the European Union - NextGenerationEU.
- ISCRA Project "AI for weather analysis and forecast" (AIWAF)

Statements and Declarations

The authors declare no competing interests.

Code availability

The code relative to the presented work is archived in the following GitHub repository. The ERA-5 dataset used in our experiments can be openly accessed at the Copernicus website.

References

[1] Surcel, M., Zawadzki, I., Yau, M.K.: A study on the scale dependence of the predictability of precipitation patterns. Journal of the Atmospheric Sciences **72**(1), 216–235 (2015). https://doi.org/10.1175/JAS-D-14-0071.1

[2] Ashok, S.P., Pekkat, S.: A systematic quantitative review on the performance of some of the recent short-term rainfall forecasting techniques. Journal of Water and Climate Change **13**(8), 3004–3029 (2022)

[3] Sun, J., Xue, M., Wilson, J.W., Zawadzki, I., Ballard, S.P., Onvlee-Hooimeyer, J., Joe, P., Barker, D.M., Li, P.-W., Golding, B., Xu, M., Pinto, J.: Use of nwp for nowcasting convective precipitation: Recent progress and challenges. Bulletin of the American Meteorological Society **95**(3), 409–426 (2014). https://doi.org/10.1175/BAMS-D-11-00263.1

[4] Tan, C., Gao, Z., Li, S.Z.: Simvp: Towards simple yet powerful spatiotemporal predictive learning. arXiv preprint arXiv:2211.12509 (2022)

[5] Tan, C., Gao, Z., Li, S., Xu, Y., Li, S.Z.: Temporal attention unit: Towards efficient spatiotemporal predictive learning. arXiv preprint arXiv:2206.12126 (2022)

[6] Ye, Y., Gao, F., Cheng, W., Liu, C., Zhang, S.: Msstnet: A multi-scale spatiotemporal prediction neural network for precipitation nowcasting. Remote Sensing **15**(1), 137 (2022)

[7] Zhou, Y., Dong, H., El Saddik, A.: Deep learning in next-frame prediction: A benchmark review. IEEE Access **8**, 69273–69283 (2020). https://doi.org/10.1109/ACCESS.2020.2987281

[8] Shi, X., Chen, Z., Wang, H., Yeung, D., Wong, W., Woo, W.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015,

December 7-12, 2015, Montreal, Quebec, Canada, pp. 802–810 (2015). https://proceedings.neurips.cc/paper/2015/hash/07563a3fe3bbe7e3ba84431ad9d055af-Abstract.html

[9] Ayzel, G., Scheffer, T., Heistermann, M.: Rainnet v1.0: a convolutional neural network for radar-based precipitation nowcasting. Geoscientific Model Development **13**(6), 2631–2644 (2020). https://doi.org/10.5194/gmd-13-2631-2020

[10] Franch, G., Nerini, D., Pendesini, M., Coviello, L., Jurman, G., Furlanello, C.: Precipitation nowcasting with orographic enhanced stacked generalization: Improving deep learning predictions on extreme events. Atmosphere **11**(3) (2020). https://doi.org/10.3390/atmos11030267

[11] Sønderby, C.K., Espeholt, L., Heek, J., Dehghani, M., Oliver, A., Salimans, T., Agrawal, S., Hickey, J., Kalchbrenner, N.: Metnet: A neural weather model for precipitation forecasting. arXiv preprint arXiv:2003.12140 (2020)

[12] Adewoyin, R.A., Dueben, P., Watson, P., He, Y., Dutta, R.: TRU-NET: a deep learning approach to high resolution prediction of rainfall. Mach. Learn. **110**(8), 2035–2062 (2021). https://doi.org/10.1007/s10994-021-06022-6

[13] Espeholt, L., Agrawal, S., Sønderby, C., Kumar, M., Heek, J., Bromberg, C., Gazen, C., Carver, R., Andrychowicz, M., Hickey, J., *et al.*: Deep learning for twelve hour precipitation forecasts. Nature communications **13**(1), 5145 (2022)

[14] Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., Tian, Q.: Pangu-weather: A 3d high-resolution model for fast and accurate global weather forecast. CoRR **abs/2211.02556** (2022) https://arxiv.org/abs/2211.02556. https://doi.org/10.48550/arXiv.2211.02556

[15] Hatanaka, Y., Glaser, Y., Galgon, G., Torri, G., Sadowski, P.: Diffusion models for high-resolution solar forecasts. ArXiv **abs/2302.00170** (2023)

[16] Oussidi, A., Elhassouny, A.: Deep generative models: Survey. In: 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), pp. 1–8 (2018). https://doi.org/10.1109/ISACV.2018.8354080

[17] Ruthotto, L., Haber, E.: An introduction to deep generative modeling. GAMM-Mitteilungen **44**(2), 202100008 (2021)

[18] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pp. 2672–2680 (2014). https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html

[19] Jabbar, A., Li, X., Omar, B.: A Survey on Generative Adversarial Networks: Variants, Applications, and Training (2020). https://doi.org/10.48550/ARXIV.2006.05132. https://arxiv.org/abs/2006.05132

[20] Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014. JMLR Workshop and Conference Proceedings, vol. 32, pp. 1278–1286 (2014). JMLR.org. http://jmlr.org/proceedings/papers/v32/rezende14.html

[21] Kingma, D.P., Welling, M.: An introduction to variational autoencoders. Found. Trends Mach. Learn. **12**(4), 307–392 (2019). https://doi.org/10.1561/2200000056

[22] Asperti, A., Evangelista, D., Piccolomini, E.L.: A survey on variational autoencoders from a green AI perspective. SN Comput. Sci. **2**(4), 301 (2021). https://doi.org/10.1007/s42979-021-00702-9

[23] Asperti, A., Tonelli, V.: Comparing the latent space of generative models. Neural Computing & Applications **To appear** (2022). https://doi.org/10.1007/s00521-022-07890-2

[24] Bau, D., Zhu, J.-Y., Wulff, J., Peebles, W., Zhou, B., Strobelt, H., Torralba, A.: Seeing what a gan cannot generate, 4501–4510 (2019). https://doi.org/10.1109/ICCV.2019.00460

[25] Leinonen, J., Nerini, D., Berne, A.: Stochastic super-resolution for downscaling time-evolving atmospheric fields with a generative adversarial network. IEEE Trans. Geosci. Remote. Sens. **59**(9), 7211–7223 (2021). https://doi.org/10.1109/TGRS.2020.3032790

[26] Price, I., Rasp, S.: Increasing the accuracy and resolution of precipitation forecasts using deep generative models. In: International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event, pp. 10555–10571 (2022). https://proceedings.mlr.press/v151/price22a.html

[27] Harris, L., McRae, A.T.T., Chantry, M., Dueben, P.D., Palmer, T.N.: A generative deep learning approach to stochastic downscaling of precipitation forecasts. Journal of Advances in Modeling Earth Systems **14**(10), 2022–003120 (2022) https://arxiv.org/abs/https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2022MS003120. https://doi.org/10.1029/2022MS003120

[28] Hayatbini, N., Kong, B., Hsu, K.-l., Nguyen, P., Sorooshian, S., Stephens, G., Fowlkes, C., Nemani, R., Ganguly, S.: Conditional generative adversarial networks (cgans) for near real-time precipitation estimation from multispectral goes-16 satellite imageries—persiann-cgan. Remote Sensing **11**(19) (2019). https://doi.org/10.3390/rs11192193

[29] Wang, C., Tang, G., Gentine, P.: Precipgan: Merging microwave and infrared data for satellite precipitation estimation using generative adversarial network. Geophysical Research Letters **48**(5), 2020–092032 (2021) https://arxiv.org/abs/https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2020GL092032. https://doi.org/10.1029/2020GL092032

[30] Scher, S., Peßenteiner, S.: Technical note: Temporal disaggregation of spatial rainfall fields with generative adversarial networks. Hydrology and Earth System Sciences **25**(6), 3207–3225 (2021). https://doi.org/10.5194/hess-25-3207-2021

[31] Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., Fitzsimons, M., Athanassiadou, M., Kashem, S., Madge, S., *et al.*: Skilful precipitation nowcasting using deep generative models of radar. Nature **597**(7878), 672–677 (2021)

[32] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual (2020). https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html

[33] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical Text-Conditional Image Generation with CLIP Latents. arXiv (2022). https://arxiv.org/abs/2204.06125

[34] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic text-to-image diffusion models with deep language understanding. CoRR **abs/2205.11487** (2022) https://arxiv.org/abs/2205.11487. https://doi.org/10.48550/arXiv.2205.11487

[35] Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., et al.: Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 (2022)

[36] Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. arXiv preprint arXiv:2204.03458 (2022)

[37] Dhariwal, P., Nichol, A.Q.: Diffusion models beat gans on image synthesis. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, Virtual, pp. 8780–8794 (2021). https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html

[38] Asperti, A., Evangelista, D., Marro, S., Merizzi, F.: Image embedding for denoising generative models. Artificial Intelligence Review **To appear** (2023). https://doi.org/10.48550/arXiv.2301.07485

[39] Harris, L., McRae, A.T., Chantry, M., Dueben, P.D., Palmer, T.N.: A generative deep learning approach to stochastic downscaling of precipitation forecasts. Journal of Advances in Modeling Earth Systems **14**(10), 2022–003120 (2022)

[40] Leinonen, J., Hamann, U., Nerini, D., Germann, U., Franch, G.: Latent diffusion models for generative precipitation nowcasting with accurate uncertainty quantification. CoRR **abs/2304.12891** (2023) https://arxiv.org/abs/2304.12891. https://doi.org/10.48550/arXiv.2304.12891

[41] Germann, U., Galli, G., Boscacci, M., Bolliger, M.: Radar precipitation measurement in a mountainous region. Quarterly Journal of the Royal Meteorological Society **132**(618), 1669–1692 (2006) https://arxiv.org/abs/https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1256/qj.05.190. https://doi.org/10.1256/qj.05.190

[42] Willemse, S., Furger, M.: From weather observations to atmospheric and climate sciences in switzerland: Celebrating 100 years of the swiss society for meteorology. chapter 9 (2016)

[43] Stephan, K., Klink, S., Schraff, C.: Assimilation of radar-derived rain rates into the convective-scale model cosmo-de at dwd. Quarterly Journal of the Royal Meteorological Society **134**(634), 1315–1326 (2008) https://arxiv.org/abs/https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/qj.269. https://doi.org/10.1002/qj.269

[44] Kaparakis, C., Mehrkanoon, S.: Wf-unet: Weather fusion unet for precipitation nowcasting. CoRR **abs/2302.04102** (2023) https://arxiv.org/abs/2302.04102. https://doi.org/10.48550/arXiv.2302.04102

[45] Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., De Chiara, G., Dahlgren, P., Dee, D., Diamantakis, M., Dragani, R., Flemming, J., Forbes, R., Fuentes, M., Geer, A., Haimberger, L., Healy, S., Hogan, R.J., Hólm, E., Janisková, M., Keeley, S., Laloyaux, P., Lopez, P., Lupu, C., Radnoti, G., de Rosnay, P., Rozum, I., Vamborg, F., Villaume, S., Thépaut, J.-N.: The era5 global reanalysis. Quarterly Journal of the Royal Meteorological Society **146**(730), 1999–2049 (2020) https://arxiv.org/abs/https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/qj.3803. https://doi.org/10.1002/qj.3803

[46] Song, J., Meng, C., Ermon, S.: Denoising Diffusion Implicit Models. arXiv e-prints, 2010–02502 (2020) https://arxiv.org/abs/2010.02502. https://doi.org/10.48550/arXiv.2010.02502

[47] Sohl-Dickstein, J., Weiss, E.A., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics **37**, 2256–2265 (2015)

[48] Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August

2017. Proceedings of Machine Learning Research, vol. 70, pp. 2642–2651 (2017). http://proceedings.mlr.press/v70/odena17a.html

[49] Ho, J., Salimans, T.: Classifier-free diffusion guidance. CoRR **abs/2207.12598** (2022) https://arxiv.org/abs/2207.12598. https://doi.org/10.48550/arXiv.2207.12598

[50] Fernández, J.G., Mehrkanoon, S.: Broad-UNet: Multi-scale feature learning for nowcasting tasks. Neural Networks **144**, 419–427 (2021). https://doi.org/10.1016/j.neunet.2021.08.036

[51] Trebing, K., Stanczyk, T., Mehrkanoon, S.: SmaAt-UNet: Precipitation Nowcasting using a Small Attention-UNet Architecture (2021)

[52] Gurrola-Ramos, J., Dalmau, O., Alarcón, T.E.: A residual dense u-net neural network for image denoising. IEEE Access **9**, 31742–31754 (2021). https://doi.org/10.1109/ACCESS.2021.3061062

[53] Lee, S., Negishi, M., Urakubo, H., Kasai, H., Ishii, S.: Mu-net: Multi-scale u-net for two-photon microscopy image denoising and restoration. Neural Networks **125**, 92–103 (2020). https://doi.org/10.1016/j.neunet.2020.01.026

[54] Heinrich, M.P., Stille, M., Buzug, T.M.: Residual u-net convolutional neural network architecture for low-dose ct denoising. Current Directions in Biomedical Engineering **4**(1), 297–300 (2018)

[55] Komatsu, R., Gonsalves, T.: Comparing u-net based models for denoising color images. AI **1**(4), 465–486 (2020)

[56] Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)

[57] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention, pp. 234–241 (2015). Springer

[58] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp. 5998–6008 (2017). https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[59] Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)

[60] Vannitsem, S., Wilks, D.S., Messner, J.: Statistical postprocessing of ensemble forecasts (2018)

[61] Gneiting, T., Raftery, A.E., Westveld, A.H., Goldman, T.: Calibrated probabilistic forecasting using ensemble model output statistics and minimum crps estimation. Monthly Weather Review **133**(5), 1098–1118 (2005)

[62] Henzi, A., Ziegel, J.F., Gneiting, T.: Isotonic distributional regression. Journal of the Royal Statistical Society Series B: Statistical Methodology **83**(5), 963–993 (2021)

[63] Taillardat, M., Mestre, O., Zamo, M., Naveau, P.: Calibrated ensemble forecasts using quantile regression forests and ensemble model output statistics. Monthly Weather Review **144**(6), 2375–2393 (2016)

[64] Messner, J.W., Mayr, G.J., Zeileis, A.: Nonhomogeneous boosting for predictor selection in ensemble postprocessing. Monthly Weather Review **145**(1), 137–147 (2017)

[65] Ashkboos, S., Huang, L., Dryden, N., Ben-Nun, T., Dueben, P., Gianinazzi, L., Kummer, L., Hoefler, T.: ENS-10: A dataset for post-processing ensemble weather forecast. CoRR **abs/2206.14786** (2022) https://arxiv.org/abs/2206.14786. https://doi.org/10.48550/arXiv.2206.14786

[66] Schulz, B., Lerch, S.: Machine learning methods for postprocessing ensemble forecasts of wind gusts: A systematic comparison. Monthly Weather Review **150**(1), 235–257 (2022). https://doi.org/10.1175/MWR-D-21-0150.1

[67] Asperti, A., Evangelista, D., Marzolla, M.: Dissecting flops along input dimensions for greenai cost estimations. In: 7th International Conference on Machine Learning, Optimization & Data Science, Grasmere, Lake District, England – UK, October 5-8 2021. Springer International Publishing, pp. 86–100 (2022). https://doi.org/10.1007/978-3-030-95470-3_7

[68] Bowler, N., Pierce, C., Seed, A.: Steps: A probabilistic precipitation forecasting scheme which merges an extrapolation nowcast with downscaled nwp. Quarterly Journal of the Royal Meteorological Society **132**, 2127–2155 (2007). https://doi.org/10.1256/qj.04.100

[69] Seed, A., Pierce, C., Norman, K.: Formulation and evaluation of a scale decomposition-based stochastic precipitation nowcast scheme. Water Resources Research **49** (2013). https://doi.org/10.1002/wrcr.20536

[70] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)

[71] Leinonen, J., Hamann, U., Nerini, D., Germann, U., Franch, G.: Latent diffusion models for generative precipitation nowcasting with accurate uncertainty quantification. arXiv e-prints, 2304–12891 (2023) https://arxiv.org/abs/2304.12891 [physics.ao-ph]. https://doi.org/10.48550/arXiv.2304.12891

[72] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, pp. 10674–10685 (2022). https://doi.org/10.1109/CVPR52688.2022.01042. https://doi.org/10.1109/CVPR52688.2022.01042

[73] Guibas, J., Mardani, M., Li, Z., Tao, A., Anandkumar, A., Catanzaro, B.: Adaptive Fourier Neural Operators: Efficient Token Mixers for Transformers. arXiv e-prints, 2111–13587 (2021) https://arxiv.org/abs/2111.13587 [cs.CV]. https://doi.org/10.48550/arXiv.2111.13587

[74] Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., Hassanzadeh, P., Kashinath, K., Anandkumar, A.: FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators. arXiv e-prints, 2202–11214 (2022) https://arxiv.org/abs/2202.11214 [physics.ao-ph]. https://doi.org/10.48550/arXiv.2202.11214

[75] García Fernández, J., Alaoui Abdellaoui, I., Mehrkanoon, S.: Deep coastal sea elements forecasting using U-Net based models. arXiv e-prints, 2011–03303 (2020) https://arxiv.org/abs/2011.03303 [cs.LG]. https://doi.org/10.48550/arXiv.2011.03303

[76] Ayzel, G., Heistermann, M., Sorokin, A., Nikitin, O., Lukyanova, O.: All convolutional neural networks for radar-based precipitation nowcasting. Procedia Computer Science **150**, 186–192 (2019). https://doi.org/10.1016/j.procs.2019.02.036. Proceedings of the 13th International Symposium "Intelligent Systems 2018" (INTELS'18), 22-24 October, 2018, St. Petersburg, Russia

[77] Bromberg, C.L., Gazen, C., Hickey, J.J., Burge, J., Barrington, L., Agrawal, S.: Machine learning for precipitation nowcasting from radar images, p. 4 (2019)

[78] Tuel, A., Martius, O.: The influence of modes of climate variability on the sub-seasonal temporal clustering of extreme precipitation. iScience **25**(3), 103855 (2022). https://doi.org/10.1016/j.isci.2022.103855

[79] Le, P., Randerson, J., Willett, R., Wright, S., Smyth, P., Guilloteau, C., Mamalakis, A., Foufoula-Georgiou, E.: Climate-driven changes in the predictability of seasonal precipitation.

Nature Communications **14** (2023). https://doi.org/10.1038/s41467-023-39463-9

[80] Ray, P.S. American Meteorological Society (1986). https://books.google.it/books?id=HDBRAAAAMAAJ

[81] Stensrud, D.J., Xue, M., Wicker, L.J., Kelleher, K.E., Foster, M.P., Schaefer, J.T., Schneider, R.S., Benjamin, S.G., Weygandt, S.S., Ferree, J.T., Tuell, J.P.: Convective-scale warn-on-forecast system: A vision for 2020. Bulletin of the American Meteorological Society **90**(10), 1487–1500 (2009). https://doi.org/10.1175/2009BAMS2795.1

[82] Ahrens, C.D. International student edition. Cengage Learning (2006). https://books.google.it/books?id=SpGfKb23Y9QC

[83] Palmer, T.N.: The economic value of ensemble forecasts as a tool for risk assessment: From days to decades. Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography **128**(581), 747–774 (2002)