

RESEARCH ARTICLE

10.1029/2022MS003120

Key Points:

- We use generative adversarial neural networks to post-process global weather forecast model output over the UK
- We produce more realistic precipitation forecasts than the input forecast data, at 10X resolution, with excellent statistical properties
- We match or outperform a state-of-the-art pointwise downscaling scheme, while also producing spatially coherent images

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

L. Harris,
lucy.harris2@physics.ox.ac.uk

Citation:

Harris, L., McRae, A. T. T., Chantry, M., Dueben, P. D., & Palmer, T. N. (2022). A generative deep learning approach to stochastic downscaling of precipitation forecasts. *Journal of Advances in Modeling Earth Systems*, 14, e2022MS003120. <https://doi.org/10.1029/2022MS003120>

Received 4 APR 2022

Accepted 1 OCT 2022

Author Contributions:

Conceptualization: Andrew T. T. McRae, Matthew Chantry, Peter D. Dueben, Tim N. Palmer

Data curation: Lucy Harris, Andrew T. T. McRae, Matthew Chantry

Formal analysis: Lucy Harris, Andrew T. T. McRae, Matthew Chantry

Funding acquisition: Matthew Chantry, Peter D. Dueben, Tim N. Palmer

A Generative Deep Learning Approach to Stochastic Downscaling of Precipitation Forecasts

Lucy Harris¹ , Andrew T. T. McRae¹ , Matthew Chantry² , Peter D. Dueben² , and Tim N. Palmer¹

¹Department of Physics, University of Oxford, Oxford, UK, ²European Centre for Medium-Range Weather Forecasts, Reading, UK

Abstract Despite continuous improvements, precipitation forecasts are still not as accurate and reliable as those of other meteorological variables. A major contributing factor to this is that several key processes affecting precipitation distribution and intensity occur below the resolved scale of global weather models. Generative adversarial networks (GANs) have been demonstrated by the computer vision community to be successful at super-resolution problems, that is, learning to add fine-scale structure to coarse images. Leinonen et al. (2020, <https://doi.org/10.1109/TGRS.2020.3032790>) previously applied a GAN to produce ensembles of reconstructed high-resolution atmospheric fields, given coarsened input data. In this paper, we demonstrate this approach can be extended to the more challenging problem of increasing the accuracy and resolution of comparatively low-resolution input from a weather forecasting model, using high-resolution radar measurements as a “ground truth.” The neural network must learn to add resolution and structure whilst accounting for non-negligible forecast error. We show that GANs and VAE-GANs can match the statistical properties of state-of-the-art pointwise post-processing methods whilst creating high-resolution, spatially coherent precipitation maps. Our model compares favorably to the best existing downscaling methods in both pixel-wise and pooled CRPS scores, power spectrum information and rank histograms (used to assess calibration). We test our models and show that they perform in a range of scenarios, including heavy rainfall.

Plain Language Summary The processes that lead to precipitation (rainfall) happen on a very small scale. Weather forecast computer models work on much larger scales, so rainfall is often poorly predicted. In this paper, we develop a method that enhances the resolution of rainfall forecasts by a factor of 10, and makes the forecasts more accurate. We generate many samples of what the rainfall pattern could be, which gives an idea of the uncertainty in the forecast. Our method is based on machine learning and neural networks, which means that we use many past examples of weather forecasts, together with the rainfall that actually happened, and our method “automatically” learns how the forecasts can be improved. We use an existing idea called “Generative Adversarial Networks,” which has been used very successfully in image-related tasks, such as producing realistic higher-resolution images from low-resolution ones. Our task is similar to producing a high-resolution image from a low-resolution one, hence this approach is promising. Our method outperforms a variety of existing approaches, and even produces good predictions for the most extreme rainfall situations in our data set. These are the scenarios that cause the most real-world disruption, the most useful events to produce good forecasts for.

1. Introduction

Weather prediction and climate models are constantly evolving, and are generally considered to perform well for most applications. However, it is a well-recognized problem that precipitation events are imperfectly predicted (Applequist et al., 2002; Berrocal et al., 2008; Gascón et al., 2018; Sha et al., 2020). This is in part due to the low spatial resolution of the outputs of most models: global weather and climate models are produced on a much larger spatial scale than is typically required to accurately predict the finer structures and extremes of rainfall events (Adewoyin et al., 2021). Limitations of computational resources, numerical stability, and knowledge of initial conditions lead to constraints on model resolution such that most global numerical forecast models operate at roughly 10–80 km grid spacings, and are consequently only capable of resolving large-scale weather phenomena, as well as a limited representation of small mesoscale atmospheric processes, topography, and land-sea distribution (D’Onofrio et al., 2014; Feser et al., 2011). When a major weather event hits some part of the world, devastating the local population, it is only days later that emergency relief is distributed (Palmer, 2020a, 2020b).

Investigation: Lucy Harris, Andrew T. T. McRae, Matthew Chantry

Methodology: Lucy Harris, Andrew T. T. McRae, Matthew Chantry, Peter D. Dueben

Project Administration: Matthew Chantry, Peter D. Dueben, Tim N. Palmer

Resources: Andrew T. T. McRae, Matthew Chantry, Peter D. Dueben, Tim N. Palmer

Software: Lucy Harris, Andrew T. T. McRae, Matthew Chantry

Supervision: Matthew Chantry, Peter D. Dueben, Tim N. Palmer

Validation: Lucy Harris, Andrew T. T. McRae, Matthew Chantry

Visualization: Lucy Harris

Writing – original draft: Lucy Harris, Andrew T. T. McRae, Matthew Chantry

Writing – review & editing: Lucy Harris, Andrew T. T. McRae, Matthew Chantry, Peter D. Dueben, Tim N. Palmer

The direct application of weather and climate model outputs to precipitation impact assessment is therefore inadequate (Yu et al., 2016), particularly for extreme rainfall and situations with significant small-scale variability, for example, in the presence of heterogeneous orography and along coastlines.

In weather and climate sciences, downscaling refers to an operation that infers high-resolution information from lower-resolution data. Confusingly, downsampling (upsampling) in machine learning refers to a reduction (increase) in the image resolution. In this paper we will discuss downscaling in a weather-related context, and upsampling in a computer vision or machine learning context. Downscaling is particularly important in precipitation forecasting: the intensity of precipitation can vary considerably over short spatial scales (1 km or less), which is much lower than the typical resolution of global weather models. Increasing the resolution of precipitation forecasts is essential for assessing the potential impacts, particularly for extreme rainfall scenarios. Increasingly, stochastic downscaling techniques are applied to generate ensembles of possible small-scale rainfall fields from an initial large-scale distribution, as a way to introduce rainfall variability at scales not resolved by physical models, since full, high-resolution, deterministic models are computationally intractable (D'Onofrio et al., 2014). Palmer (2020a, 2020b) advocate for the use of stochastic neural network approaches to post-process and downscale global forecast model output, perhaps in place of traditional limited-area models. In stochastic downscaling, the goal is to produce an ensemble of possible realizations where the small-scale fields are consistent with the large scale features of the low-resolution data, as well as any smaller-scale information, such as the terrain geometry or land-sea distribution. Downscaling is inherently an under-determined problem, where one low-resolution forecast state could be valid for a multitude of high-resolution truths. This low-resolution state will generally contain errors when compared to a coarsened version of the truth data. By employing a stochastic method we can sample these high-resolution states to capture the uncertainty of both the mapping between data sources and the downscaling. We can condition this mapping further by including additional model fields and surface descriptors.

Downscaling precipitation using convolutional neural networks is a very active area of research. Many authors have approached the problem as a pure super-resolution task by coarsening their “truth” data and inputting this to their model (sometimes alongside other fields), then trying to retrieve the lost resolution. Papers that take this approach include Sha et al. (2020), Wang et al. (2021), and Kumar et al. (2021). However, we argue that this is not sufficient to tackle the full downscaling problem, since it does not account for the inevitable errors in the input forecast data. Two papers that use independent input and truth data sets are Huang (2020) and Adewoyin et al. (2021). However, both of these are motivated by climate models rather than weather prediction and so operate at much coarser scales in space and time than we do. This changes the flavor of the problem substantially—for example, both papers prioritize standard metrics like RMSE, which is inappropriate on higher resolutions and shorter timescales (Rossa et al., 2008). Hess and Boers (2022) uses independent data sets, and has a particular focus on heavy rainfall events, but does not increase resolution. Finally, many authors have used convolutional neural networks for nowcasting: forecasting precipitation events over short lead times (typically 0–6 hr). This problem differs from the downscaling task examined here, with a focus on evolving fields forward in time instead of enhancing and increasing resolution of input data. Nevertheless, many network architecture elements are shared across these domains. Recent work in this area includes Shi et al. (2015), Agrawal et al. (2019), Søndersby et al. (2020), Ravuri et al. (2021), Kloczek et al. (2021), and Espeholt et al. (2021).

In digital image processing, super-resolution refers to enhancing the spatial resolution of an image by estimating a high resolution image from its low-resolution counterpart. This has clear parallels with the downscaling problem from weather and climate science. Super-resolution is a highly challenging task, receives substantial attention within the computer vision research community, and has a wide range of applications. Recent developments in this field have led to the application of convolutional neural networks (CNNs), and subsequently, generative adversarial networks (GANs) (Goodfellow et al., 2014) to super-resolution problems (Dong et al., 2015; Ledig et al., 2017; Lin et al., 2017). The purpose of a GAN model is to generate realistic artificial samples similar to those encountered during training. GANs differ from typical neural network approaches—in place of a standard “loss function,” a second network (the discriminator) is used to evaluate generated samples. The generator network is hence trained to produce outputs that the discriminator considers to be realistic, while the discriminator is trained to better differentiate between real and artificial data. This approach has found great success in super-resolution applications.

Generative adversarial network approaches have started to appear in post-processing/downscaling and forecasting/nowcasting applications. Bihlo (2020) produced 24-hr large-scale predictions, trained on ERA5 reanalysis

data. Promising results were obtained for 500 hPa geopotential height and 2 m temperature, but not for precipitation. Watson et al. (2020) performs precipitation downscaling, trained to map between different configurations of the Weather Research and Forecasting Model (WRF). The results are promising, although only a preliminary analysis is presented. Ravuri et al. (2021) successfully tackled the precipitation nowcasting problem, producing high-resolution 90-min forecasts over the UK. Gong et al. (2022) forecasts the evolution of 2 m temperature over 12 hr, trained on ERA5 data, using an existing adversarial video-prediction architecture.

Previously, Leinonen et al. (2020) successfully applied a GAN to stochastically downscale time-series of atmospheric fields, including precipitation. However, this took the pure super-resolution approach of first coarsening the high-resolution “truth” data and then recovering the lost resolution. The absence of future radar truth images means that any application of Leinonen’s model would have to infer from a future forecast model state, which is somewhat different to the task for which it was trained. In this paper, we work on an extension of this problem. We do not just learn the mapping from coarse-to fine-resolution representations of the same data. Instead, our models learn the mapping from (multiple) low-resolution atmospheric fields, originating from a weather forecast model, to high-resolution “truth” radar data. Thus, we are aiming to both increase resolution of the original forecast and provide error correction in a probabilistic sense. The neural networks are also supplied with high-resolution orography data and a land-sea mask, which are expected to affect local precipitation due to physical principles (Holden et al., 2011). We are therefore tackling the complete downscaling problem: using the predictive power of atmospheric model fields and surface properties to match an observation of Earth’s weather. Following the excellent performance of the Leinonen approach we closely follow their model architecture. However, due to computational constraints we have removed the time-series aspect of Leinonen’s approach.

Shortly before completion of this work, Price and Rasp (2022) appeared in the literature, which builds upon Leinonen’s GAN model in a similar way to us. Like us, they map low-resolution weather forecast data to a higher-resolution precipitation truth data set. Their downscaling factor is comparable to ours, but they work at coarser resolutions in space and time. They make several different choices to us regarding network inputs and training, and an optimal solution may well combine strengths from both approaches.

2. Data

We trained our model to map hourly data from the Integrated Forecast System (IFS) to hourly accumulated rainfall based on the NIMROD radar network (Met Office, 2003). Our domain of interest covers latitudes 49.5° – 59° and longitudes -7.5° – 2° , covering mainland UK.

2.1. IFS Data

Our input data is the ECMWF’s IFS operational forecast data set, using years 2016–2020. During training we use 7–17 hr lead time forecasts, initialized at 00Z and 12Z. Earlier lead times are discarded to ensure any artifacts from data assimilation do not affect training. Later lead times are discarded as the chaotic nature of the atmosphere means the predicted cloud locations within the IFS will be increasingly poorly aligned with real world observations. We did not test the sensitivity of these two lead time thresholds, but we later evaluate the model on lead times out to 72 hr and find the model performs well despite only being trained on short-term forecasts.

From the IFS model we use 9 fields:

- Total precipitation
- Convective precipitation
- Surface pressure
- TOA incident solar radiation
- Convective available potential energy
- Total column cloud liquid water
- Total column water vapor
- u & v (horizontal wind) velocities at 700 hPa

The choice of these fields was motivated by the ecPoint model (Hewson & Pilloso, 2021), and domain knowledge. IFS data is linearly interpolated to a 0.1° grid (approximately 10 km), resulting in images of size 94×94 pixels. To normalize the precipitation fields in the IFS data (total and convective precipitation) for input into the

neural networks, we use the transformation $\log_{10}(1 + x)$ on the mm/hr rate. The surface pressure field is normalized by subtracting the mean and dividing by the standard deviation, where these values are scalars, calculated from all grid points in 2018. Each of the other fields are normalized by calculating the (absolute) maximum value observed in 2018 (across all grid points) and dividing by this value. Winds u & v are normalized independently from one another. The inputs to the neural network are hence all $\mathcal{O}(1)$.

2.2. NIMROD Data

As a “truth” data set, we use the 1 km Resolution UK Composite Rainfall Data from the Met Office NIMROD System (Met Office, 2003). This system delivers radar-derived precipitation maps every 5 min, covering 2004 to present day. As with the IFS we use 2016–2020 data and aggregate to hourly precipitation. Calendar days with more than 30 min of missing data were removed from the data set. On average this results in 330 days per year, or approximately 8,000 hr. For ease of grid alignment with the IFS the data is re-gridded to a 0.01° grid, resulting in images of size 940×940 pixels. We again use a $\log_{10}(1 + x)$ transformation for the NIMROD precipitation.

The NIMROD data set itself naturally has inherent errors, and even contains obvious artifacts resulting from the radar system. However, we believe the NIMROD data is different enough to the IFS input, and close enough to the genuine “truth,” that training a successful model on the NIMROD data is of equivalent difficulty to training a model on any more accurate precipitation data set that may be available in the future. Furthermore, even though the data is imperfect, the trained models will still provide significant value over the IFS input. As a result, we make no further steps of data cleaning or processing to the NIMROD data set.

2.3. Geographic Data

To improve model performance we augment our model input data with high resolution surface geopotential and land-sea mask data, which depend only on location and do not vary with time and date. These may help the network add meaningful information on length scales smaller than the input data. These fields are derived from 1.25 km input data (originally generated for high resolution IFS simulations) and are re-gridded to the same 0.01° grid as the NIMROD data set. The surface geopotential is normalized by dividing by the global maximum value. Before this, values less than $5 \text{ m}^2 \text{ s}^{-2}$ are clipped to this value. This is done to remove artifacts stemming from the spectral origin of our data. The land-sea mask already takes fractional values between 0 (no land in grid box) to 1 (grid box only comprised of land).

2.4. Data Subsets

The model was trained on data from 2016 to 2018. Data from 2019 was used for validation, and data from 2020 was held out for final testing. All quantitative evaluation in this paper is performed solely on 2020 data. Some interesting synoptical situations from 2019 have also been included as case studies.

Contrary to popular opinion, it is not raining in the UK for the overwhelming majority of the time. We were therefore concerned that training the model on randomly-sampled input data would cause significant under-prediction of rainfall during high-rainfall events. Furthermore, although we could use full-sized low- and high-resolution images during inference (94×94 and 940×940 , respectively), we did not have the computational resources to use such large images during model training. The data across the UK were therefore split into smaller sub-images of 20×20 (low-resolution) and 200×200 (high-resolution), by randomly sampling patches from the full-sized images. Each sub-image was scored on “how rainy” it was in that image and categorized into one of four bins, depending on what fraction of pixels contained rainfall ($>0.1 \text{ mm/hr}$) – 0%–25%, 25%–50%, 50%–75%, or 75%–100%. This allowed us to select the distribution with which we sample from the different bins during model training, and we treated this as a hyperparameter to be optimized. The results of varying the distribution of images shown to the network are discussed in the appendix. We remark that Ravuri et al. (2021) also increased the prevalence of rainy images in their training data, although their weighting was based on both spatial coverage and intensity.

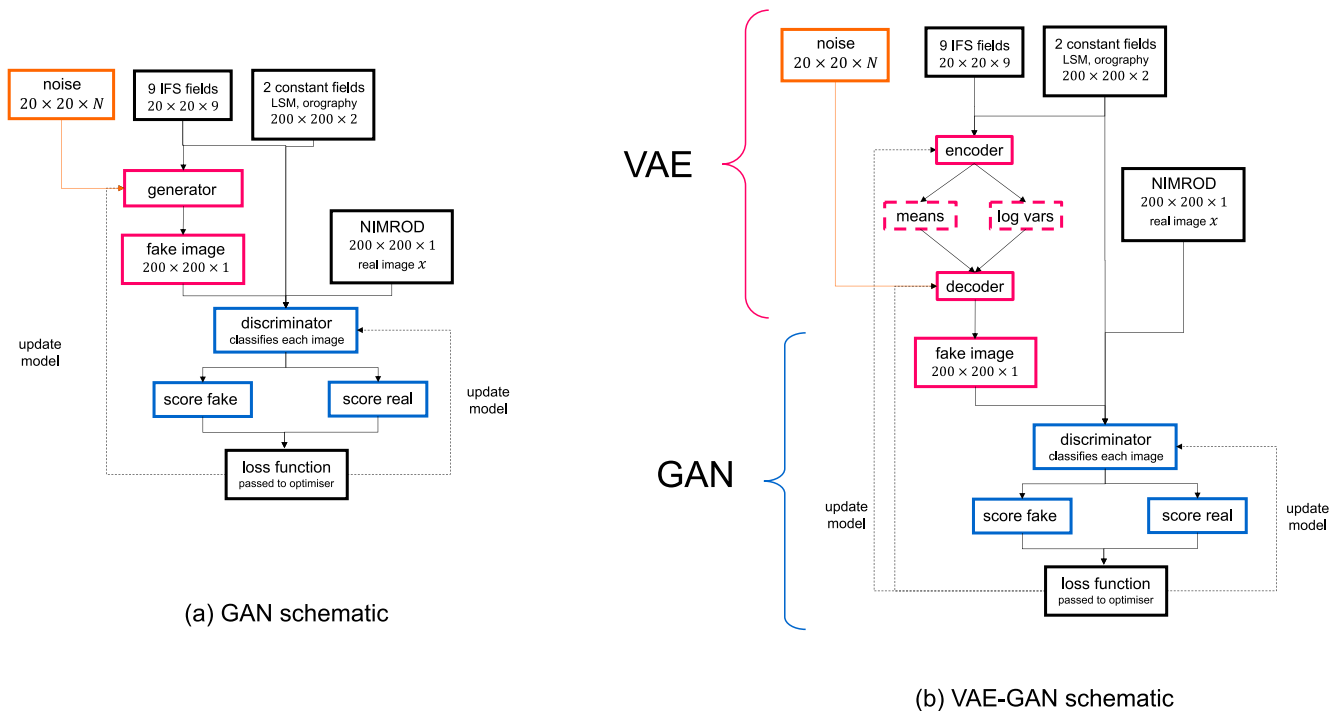


Figure 1. Schematic of the information flow through (a) the conditional generative adversarial network (GAN) model and (b) the conditional variational auto-encoder-generative adversarial network (VAE-GAN) model.

3. Methods

We use two generative deep learning approaches; both post-process lower-resolution atmospheric field forecast data and aim to produce well-calibrated ensembles of high-resolution precipitation forecasts.

3.1. Model 1: GAN

The first model is a conditional GAN (Mirza & Osindero, 2014), where both the generator and the discriminator are conditioned on additional information: in this case, lower-resolution atmospheric fields and full-resolution orography and land-sea mask data. The generator has an explicit noise input, which allows multiple samples to be generated for a given forecast state. The discriminator is trained to distinguish between the high-resolution predictions from the generator and corresponding “ground-truth” high-resolution rainfall data. We follow the work of Arjovsky et al. (2017) and Gulrajani et al. (2017) by using a Wasserstein-GAN with a gradient penalty to enable stable GAN training. A high-level schematic of our conditional GAN is shown in Figure 1a.

3.2. Model 2: VAE-GAN

We initially explored using a variational auto-encoder (VAE) as an alternative approach to GANs. This is a model consisting of an encoder network and a decoder network. The encoder network maps from the input to some latent space representation of the input data, encoded in the means and (log-)variances of normal random variables. The decoder network then samples from the normal distributions described by these variables, via an external noise input, and attempts to recreate the higher resolution “truth” data. This required us to define a “content loss” term that penalizes deviations between the network output and the truth data. Despite trying numerous content loss terms, results were uniformly disappointing—the resulting ensemble was greatly under-dispersive, and the predictions “blurry.”

We therefore developed a hybrid VAE-GAN model which substituted the GAN generator with a VAE. This effectively employs a full discriminator network as the VAE content loss function, and produced much sharper and better-calibrated results. A high-level schematic of our hybrid conditional VAE-GAN is shown in Figure 1b.

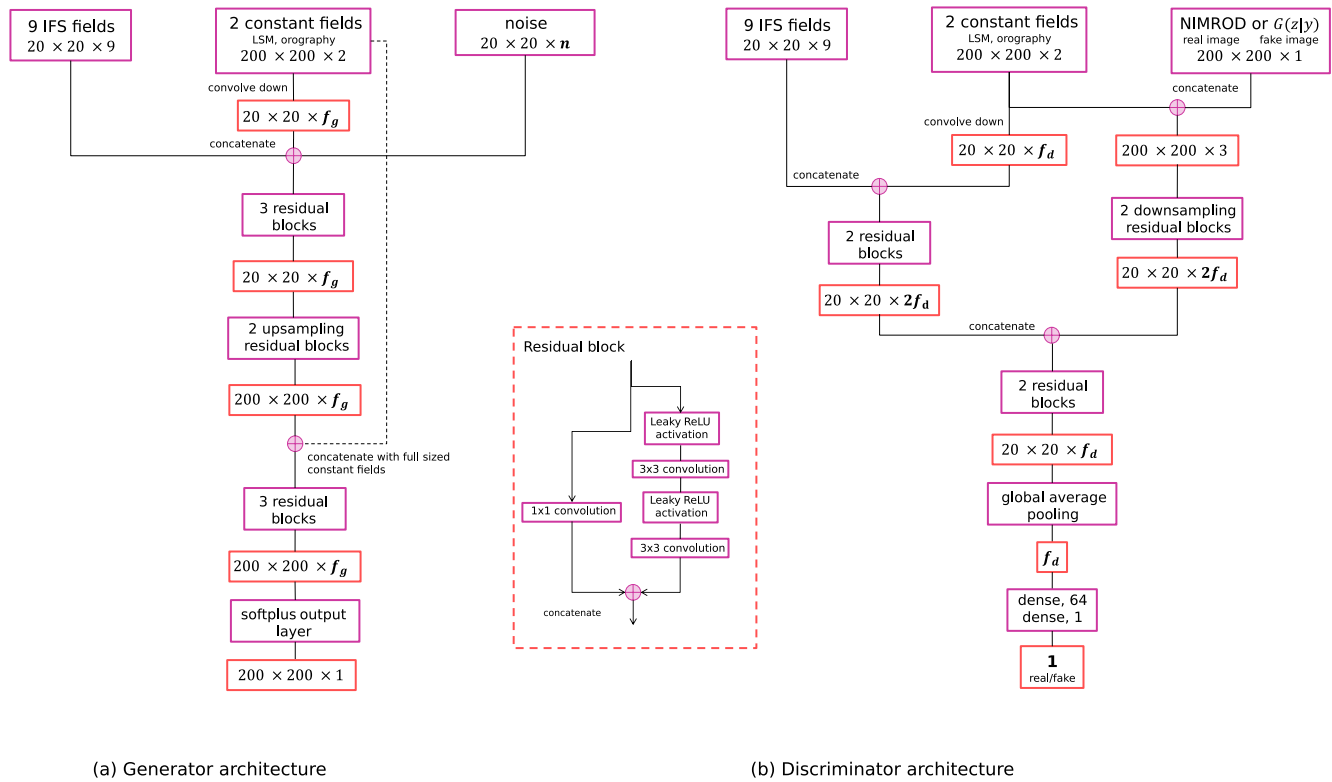


Figure 2. Network architecture for the conditional generative adversarial network (GAN): (a) the generator model and (b) the discriminator model.

3.3. Model Architecture

The generator of the GAN, encoder and decoder of the VAE-GAN, and discriminator are all deep, convolutional neural networks which make heavy use of residual blocks (He et al., 2015). The architecture is closely based on that used in Leinonen et al. (2020), modified for our downscaling factor of 10, and with blocks facilitating the temporal component of their problem removed. The generator networks in both models are fully convolutional, without any dense layers. This allows them to be size-agnostic, and hence we can train the network on 20×20 input images but use full-size 94×94 input images during inference. Due to this restriction, the latent variables in the VAE-GAN model will only represent local rather than global information.

The inputs to the models are:

- Low-resolution conditioning fields (weather forecasts), with dimensions $(b \times h_l \times w_l \times N_i)$,
- High-resolution geographic fields (land-sea mask and orography), with dimensions $(b \times h_h \times w_h \times 2)$,
- A noise input, with dimensions $(b \times h_l \times w_l \times n)$,

where b is the batch size, h_l and w_l are the low-resolution input image dimensions, N_i is the number of input conditioning fields (for us, typically 9 IFS variables), h_h and w_h are the high-resolution target image dimensions, and n is the number of noise channels per input image pixel. The ratio between the high- and low-resolution image dimensions is the *downscaling factor*, K ; in this paper, we use a downscaling factor $K = 10$ throughout. In the GAN generator, the number of noise channels, n is a parameter that can be varied. In the VAE-based models, there is one noise input for each latent variable, where the number of latent variables per pixel is a parameter that can be varied. We did not attempt to use a more sophisticated “conditioning stack” to further process the IID noise input, as was done in Ravuri et al. (2021).

3.4. GAN Architecture

The GAN architecture is displayed in Figure 2. The number of trainable parameters in the generator depends on the number of filters, f_g . When $f_g = 128$, the value we use throughout this paper, the generator network has

approximately 3.2 million trainable parameters. The number of trainable parameters in the discriminator depends on the number of filters, f_d . When $f_d = 512$, the value we use, the discriminator network has approximately 64 million trainable parameters. The networks were designed by assessing the overall performance of the architecture with different hyperparameter choices. Since a Wasserstein GAN can be trained to optimality (Gulrajani et al., 2017), we deliberately choose $f_d > f_g$ so that the discriminator network is more powerful than the generator network. This helps to avoid mode collapse from occurring. We were limited to a maximum value of $f_d = 512$ by the hardware available (initially, a V100 GPU with 16 GB RAM, although we later gained access to an A100 GPU). Increasing the number of channels in the generator had a much smaller impact on the model performance.

3.5. VAE-GAN Architecture

The VAE-GAN model has a similar architecture to the GAN model, with the key difference in where the noise input is passed to the model. In the VAE-GAN generator, the noise is introduced at an intermediate stage when the latent variable distributions are sampled. This is performed after three low-resolution residual blocks. A further three residual blocks are used, before upsampling occurs. We originally performed upsampling immediately after sampling from the latent variable distributions. The resulting ensemble members had overly similar large-scale features. The extra network layers are hence crucial for allowing the sampled latent variables to develop into coherent, larger-scale spatial variations. The rest of the generator and the entire discriminator are identical to the architecture used in the pure GAN model. The network architecture of the VAE-GAN generator model is shown in Figure 3. The discriminator model remains the same as before, shown in Figure 2.

We use a specific number of latent variables per pixel of the low-resolution image. The results shown in this paper use 50 latent variables per pixel. In early trials, we used far fewer, corresponding to a significant network bottleneck compared to the network width of 128 in other layers. However, the results were rather worse than the pure GAN, which does not have such a bottleneck. This led us to increase the number dramatically.

3.6. Remarks

Leaky rectified linear unit (ReLU) activations (Maas et al., 2013) with a negative slope of 0.2 are used in the residual blocks in both the generator and the discriminator. Regular ReLU activations are used in the upscaling (dimension-reducing) convolutions of the high-resolution input pathways of the discriminator. The final activation function used is a softplus layer on the output of the generator, leading to precipitation values (in the transformed variable). Using a softplus activation (instead of, e.g., a sigmoid) prevents the output from having an artificially-constrained maximum value, which we originally considered desirable. However, we found that in extreme convective scenarios, the network could produce ensemble members with unphysical values of localized precipitation, of order 1000 mm/hr. Recall we use a $\log_{10}(1 + x)$ variable transform for precipitation, hence the network output y is converted to a precipitation value $(10^y - 1)$ mm/hr. As a result, $\mathcal{O}(1)$ errors in extremes of y lead to order-of-magnitude errors in extremes of precipitation. We therefore clip values above 500 mm/hr, although this could be lowered somewhat further.

Formally, the network can only predict precipitation values in the half-open interval $(0, 500]$ mm/hr, in contrast to methods which explicitly assign a probability to zero precipitation (e.g., Vaughan et al. (2022)). However, the precipitation values can be arbitrarily small, and it would be a trivial post-processing step to flush values below some threshold to zero. The final activation function of the discriminator is linear, a standard choice in a WGAN. In terms of computational resources, evaluating the GAN or VAE-GAN generator on a single full-size image, mapping a 94×94 input to a 940×940 output, takes approximately 0.13 s per ensemble member on an NVIDIA A100 GPU.

4. Training and Validation

4.1. Training

The standard training objective for a GAN is a *minimax* game: the generator, G , tries to minimize a loss function, whilst the discriminator, D , tries to maximize it. This loss function represents the ability of the discriminator to tell a real sample from a fake one. In a Wasserstein GAN (Gulrajani et al., 2017), the loss function is constructed from the Wasserstein metric, or earth-mover distance.

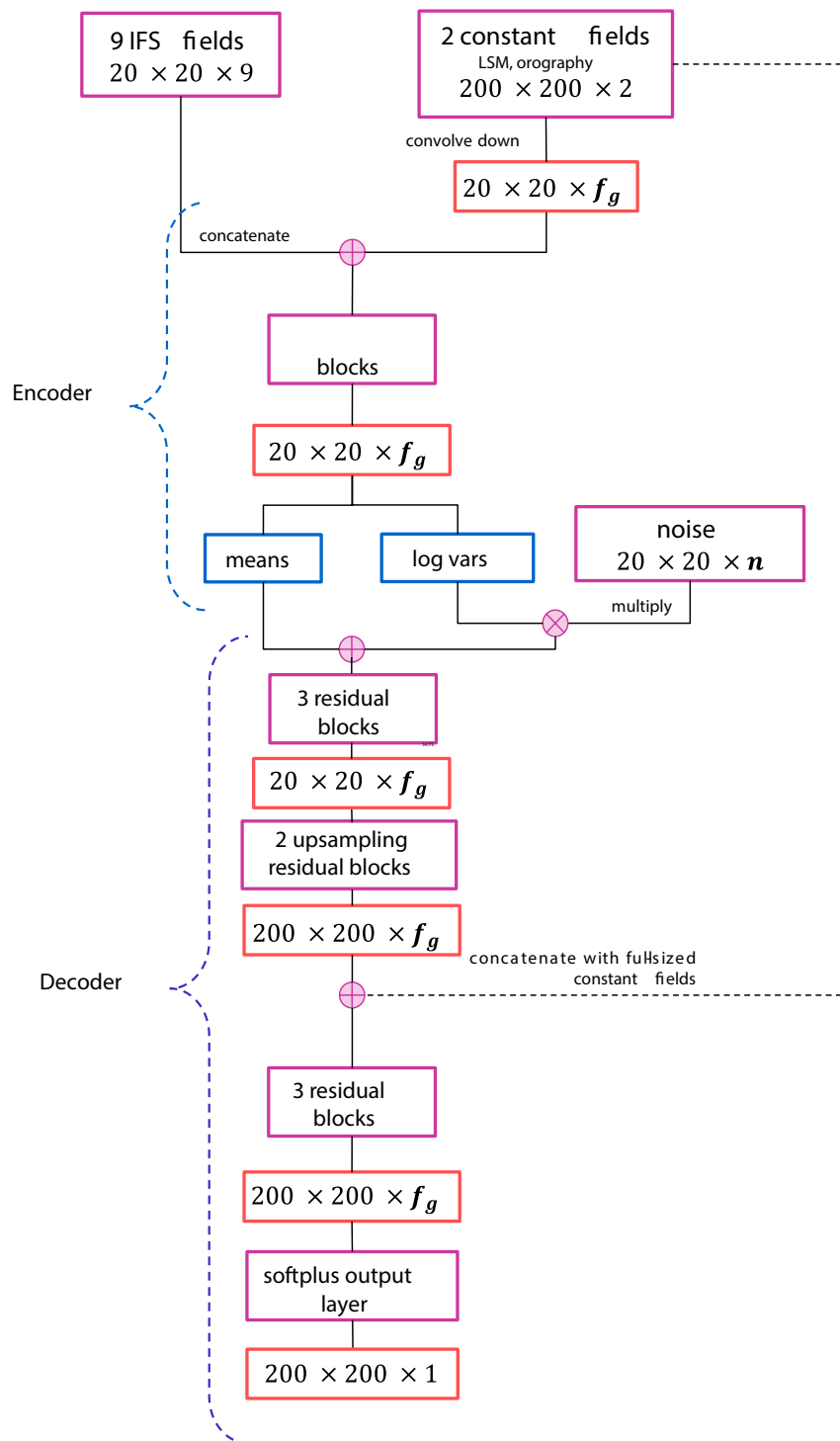


Figure 3. Network architecture for the variational auto-encoder-generative adversarial network (VAE-GAN) generator model. The discriminator model is identical to that shown in Figure 2.

In our setting, of a conditional GAN (Mirza & Osindero, 2014), both the generator and discriminator receive common inputs—the IFS data, and geographic data—which we represent by y . The GAN generator also takes in a noise input, z , while the discriminator takes in either “truth” data x_{true} , or generated data $G(z|y)$. The loss functions themselves are simple; the discriminator has a loss function:

$$\mathcal{L}_D = D(x_{\text{true}}|y) - D(G(z|y)|y) \quad (1)$$

The term $D(x_{\text{true}}|y)$ therefore represents the discriminator's “score” that the real data instance is real, while $D(G(z|y)|y)$ is the discriminator's “score” that the generated, fake instance is real. The discriminator tries to maximize this function, that is, it tries to maximize the difference between its output on real instances and its output on fake instances. The generator has loss function:

$$\mathcal{L}_G = D(G(z|y)|y) \quad (2)$$

The generator tries to maximize this function, that is, it tries to maximize the output of the discriminator for the generated fake instances. Intuitively, it tries to “trick” the discriminator into thinking the generated output is real.

Wasserstein GANs (WGANs) have various theoretical advantages over traditional GANs: They avoid problems with vanishing gradients, and the earth-mover distance is a true metric: a measure of distance in a space of probability distributions. Since it is continuous and differentiable, the discriminator can be trained to optimality (Arjovsky et al., 2017). Furthermore, in practice, WGANs are less vulnerable to getting stuck during training than traditional GANs. We follow Gulrajani et al. (2017) by using a WGAN with a gradient penalty term, as shown in Equation 3.

Motivated by Ravuri et al. (2021), we added a further “content loss” term to the generator loss function. We implement this as the mean squared error between the truth and an *ensemble mean* prediction over 8 ensemble members. This calculation is performed in the *transformed* precipitation variable. Mean squared error terms often penalize a model from making bold predictions and result in “blurry” images; this effect is far less pronounced here since the loss function is applied to an ensemble mean, rather than an individual prediction. We also experimented with a content loss term based on the pixel-wise CRPS. This produced similar results, and allowed a smaller ensemble size to be used. However, we felt we saw more instability during training, perhaps because CRPS penalizes large errors less than ensemble-mean MSE.

The loss functions for a conditional WGAN-GP, as employed in this paper, take the form:

$$\mathcal{L}_D(x_{\text{true}}, y, z; \theta_D) = \underbrace{D(x_{\text{true}}|y) - D(G(z|y)|y)}_{\text{original discriminator loss}} + \underbrace{\gamma \left(\left\| \nabla_{\hat{x}} D(\hat{x}|y) \right\|_2 - 1 \right)^2}_{\text{gradient penalty}}, \quad (3)$$

$$\mathcal{L}_G(x_{\text{true}}, y, z; \theta_G) = \underbrace{D(G(z|y)|y)}_{\text{original generator loss}} + \underbrace{\frac{\lambda}{N} \left\| x_{\text{true}} - \frac{1}{P} \sum_{i=1}^P G(z_i|y) \right\|_2^2}_{\text{content loss term}}, \quad (4)$$

where \mathcal{L}_D and \mathcal{L}_G are the loss functions for the discriminator and the generator, respectively, and θ_D and θ_G are the corresponding trainable weights, with a gradient penalty weight $\gamma = 10$, after Gulrajani et al. (2017), and content loss weight $\lambda = 1,000$ from experimentation. The samples \hat{x} , used to calculate the gradient penalty term, are randomly weighted averages of the real and generated terms:

$$\hat{x} = \epsilon x + (1 - \epsilon) G(z|y), \quad (5)$$

where ϵ is randomly sampled from a uniform distribution between 0 and 1.

For the VAE-GAN generator, the generator loss contains an additional term based on the Kullback–Leibler divergence:

$$\frac{1}{2} \left[\left(\sum_{j=1}^M \mu_j^2 + \sum_{j=1}^M \sigma_j^2 \right) - \sum_{j=1}^M (\log(\sigma_j^2) + 1) \right]. \quad (6)$$

The sums are taken over the M intermediate latent variables, whose distributions are $\{\mathcal{N}(\mu_j, \sigma_j)\}$. This term must be weighted against the original generator loss and the content loss term; we use a multiplicative factor of 10^{-5} . However, the results did not seem especially sensitive to this choice.

The generator and discriminator are trained adversarially, with the model alternating between training the discriminator for five iterations and the generator for one, after Kurach et al. (2018). The Adam optimizer (Kingma & Ba, 2014) is used for both the generator and the discriminator, with a constant learning rate of 10^{-5} for the pure GAN, and 5×10^{-6} for the VAE-GAN; we found larger learning rates resulted in unstable training. The model was trained with a batch size of 2 (limited by GPU memory) for 320,000 batches. The discriminator is trained on five times as many samples. Model weights are written to disk at 100 intermediate “checkpoints” in order to facilitate model selection, as described in Section 4.3. Training a single model took approximately 3 days, using a single NVIDIA A100 GPU.

4.2. Validation

A number of metrics are used to assess the performance of the networks. We describe them here.

4.2.1. CRPS

A commonly used distance metric in the field of weather and climate forecasting is the continuous ranked probability score (CRPS) (Gneiting & Raftery, 2007; Hersbach, 2000; Matheson & Winkler, 1976). The CRPS uses the entire ensemble of predictions to score the forecast. For each pixel in a predicted image, the CRPS is the integral of the squared difference of the cumulative distribution function (CDF) of the ensemble members, F , to the CDF of the observations. The observation CDF is a Heaviside step function H at the point $x_{\text{true},i}$. The CRPS for the pixel i is therefore:

$$\text{CRPS} = \int_{-\infty}^{\infty} (F(x') - H(x' - x_{\text{true},i}))^2 dx' \quad (7)$$

The CRPS for the entire image is the mean of the pixel-wise CRPS scores. The CRPS can therefore be understood as a generalization of the mean absolute error, and in the case of only one ensemble member it reduces to this metric.

Pixel-wise CRPS scores reward well-calibrated local forecasts, but do not promote spatially-coherent forecasts. We hence also calculate CRPS on spatially-pooled forecasts, per Ravuri et al. (2021). We use both *average-pooling* and *max-pooling*, in which we consider *average* and *maximum* values over local neighborhoods. The former can be motivated by flood forecasts, in which rainfall accumulations over larger spatial regions are relevant. The latter is perhaps relevant for extreme localized rainfall events, whose location is unlikely to be forecast precisely. We follow Ravuri et al. (2021) by using neighborhood sizes of 4×4 (stride 2) and 16×16 (stride 4).

4.2.2. Rank Histograms

These aim to assess the amount of variability in the images produced by the network. For each low-resolution sample passed into the network, we have a ground truth image and an ensemble of predictions. For each pixel in each truth image, we can therefore determine the *normalized rank* of the actual value compared to all N_p predictions: $r = \frac{N_s}{N_p}$, where N_s is the number of ensemble members below the truth. If the ensemble is perfectly calibrated, r would be uniformly distributed across the range $0 \leq r \leq 1$ when sampled enough times. The shape of the distribution of r can therefore be used as an evaluation metric to assess the variability of the generated images. We examine this distribution of r visually by plotting a histogram, after Hamill (2000). Since our networks cannot explicitly predict zero rainfall, our histograms would be distorted by the presence of zero rainfall values in the truth image. We therefore add a meteorologically-insignificant amount of noise, of order 10^{-3} mm/hr, to both the model-generated images and the ground-truth images before performing rank calculations.

Since heavy rainfall events are particularly important, we produce separate rank histogram plots that only consider events corresponding to the top 0.01% of IFS precipitation predictions within the evaluation sample. While it may sound more natural to condition on the top fraction of pixels in the “truth” NIMROD data set, we found that these pixels were generally higher than all 100 ensemble members, whether using our approach or a strong baseline method (described in Section 5.1). This is likely because there is no way to reliably predict the precise pixels that will experience heavy localized rainfall events, at least at the high spatial resolution we are working at. Our “thresholded” rank histograms are therefore conditioned on the most extreme *forecast* values.

4.2.3. Image Quality Metrics

The simplest measure of image accuracy is the root-mean-squared error. However, we found this metric to be unsuitable for assessing our model performance since we are in a regime where the well-known “double penalty problem” applies (Rossa et al., 2008). The uncertainty in small scale spatial variations is beyond what we can reliably infer from the input data, and hence predictions that forecast correct amounts of rain in slightly incorrect locations often score worse than forecasts of no rain at all. Similarly, we found that metrics like the multi-scale structural similarity index (MS-SSIM) (Z. Wang et al., 2003), which is popular in the computer vision community, were not particularly useful for our problem. We do report the ensemble mean RMSE: the root-mean-squared error of the mean of an ensemble of generated predictions.

Leinonen et al. (2020) used a log spectral distance metric (LSD) to compute a root mean square error in the 2D power spectra, in decibels (dB). However, we also found little correlation between good scores from this metric and good model predictions, perhaps because using the full 2D power spectrum is overly stringent. Instead, we compute the radially averaged power spectral density (RAPSD), which was also used in Ravuri et al. (2021). This involves calculating the 2D power spectrum, then collapsing over all angular directions (with binning) to form a 1D power spectrum. We then compute a log spectral distance of this. We are unaware of an established name for this metric, so we label this a Radially Averaged Log Spectral Distance (RALSD):

$$\text{RALSD} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(10 \log_{10} \frac{\bar{P}_{\text{true},i}}{\bar{P}_{\text{gen},i}} \right)^2} \quad (8)$$

where \bar{P}_{true} and \bar{P}_{gen} are the radially averaged power spectra of the true and generated images, respectively, and N is the number of bins. We calculate the spectra in accordance with Ruzanski and Chandrasekar (2011), using the pySTEPS implementation (Pulkkinen et al., 2019). Due to the logarithm, we found that this metric can produce distorted results in cases with very low rainfall nationwide. Since these cases are of little physical interest for our application, we exclude cases where the average rainfall over the entire image is less than 0.002 mm/hr when calculating the RALSD.

4.2.4. ROC and Precision-Recall Curves

Receiver Operating Characteristic (ROC) curves are a standard diagnostic in machine learning applications. The ROC curve assesses the skill of a binary classifier by plotting the true positive rate (sensitivity) against the false positive rate (1—specificity), across the range of probability thresholds. To construct ROC curves for a particular precipitation intensity, we make an ensemble of neural-network predictions for each forecast event. For each pixel, we look at what fraction of the ensemble members predicted rainfall above the prescribed intensity. We interpret this as the probability that our system outputs for the event taking place. Each point on the ROC curve then indicates the performance of our system when a specific probability threshold is used to separate positive predictions from negative ones. The ROC curve then indicates the performance of our system across all probability thresholds from 0 to 1, over $\mathcal{O}(10^8)$ individual predictions (i.e., each image pixel, for several hundred forecast events). We produce these curves for a range of precipitation intensities, from 0.1 mm/hr (common) to 5.0 mm/hr (rare). The ROC curve is often reduced further to a single number, the area under the curve (AUC), which would ideally be 1. This metric is also shown in the plots.

Precision-Recall curves are a closely-related diagnostic, which plot precision against sensitivity (recall). These are often considered better suited for low-probability events (Saito & Rehmsmeier, 2015), such as high rainfall intensities within our application. Precision-Recall curves for our models are shown in Supporting Information S1.

4.2.5. Fractions Skill Score

Many of the preceding metrics were point-wise, that is, defined by comparing the predictions with the truth image at a pixel-by-pixel level. The fractions skill score (FSS) (N. M. Roberts & Lean, 2008; N. Roberts, 2008) is a popular verification method that takes spatial consistency into account. For a given precipitation threshold, the prediction and truth images are binarized according to whether the rainfall is above the prescribed intensity. The neighborhood of each forecast pixel is then compared with the neighborhood of each truth pixel, based on

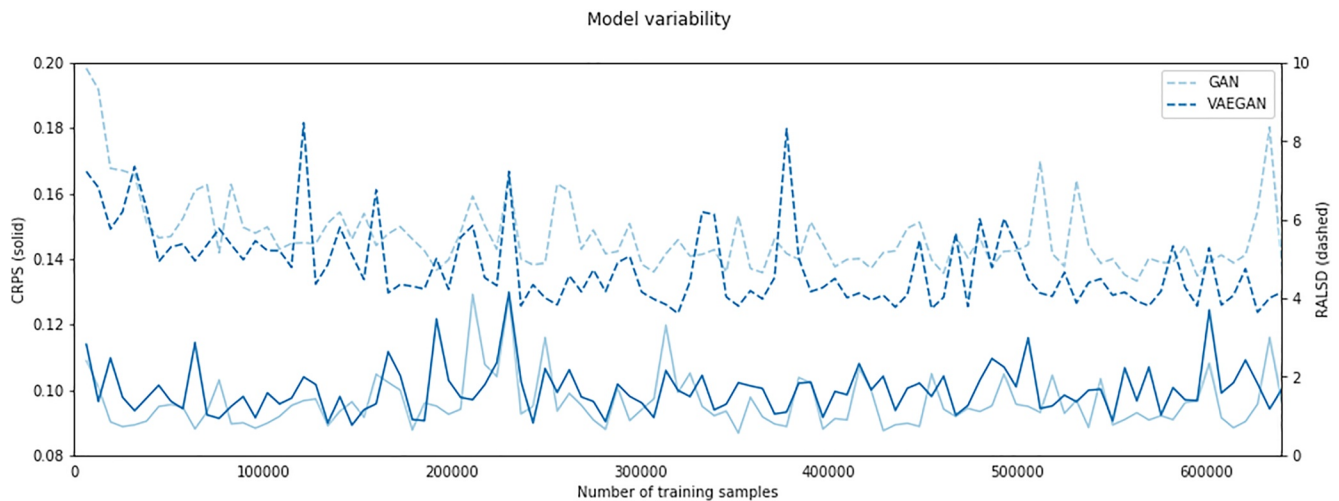


Figure 4. A representative example of the highly variable continuous ranked probability score (CRPS) and Radially Averaged Log Spectral Distance (RALSD) of generative adversarial network (GAN) and variational auto-encoder-generative adversarial network (VAE-GAN) generated samples as a function of number of generator training samples. Note that this plot is from the validation data, that is, 2019. The final one-third of these checkpoints is examined in more detail, and a model checkpoint that scores the best across multiple metrics (on the validation data) is selected for final evaluation.

the fraction of pixels meeting the criteria. A skill score is calculated from this, representing the forecast performance at a particular spatial scale. When the score exceeds a certain number, the forecast is said to have useful skill at that spatial scale.

The basic FSS compares multiple individual ensemble members with the truth sequentially. We found that this metric can be artificially inflated (at intermediate spatial scales) by forecasts with small-scale noise, so this metric should be interpreted with caution. This behavior has been observed independently (Suman Ravuri, private communication). We also use the “ensemble FSS” concept described in Duc et al. (2013), in which the binarized prediction is replaced by probabilities in $[0, 1]$, representing the proportion of ensemble members predicting rainfall above the prescribed threshold. This metric does not seem to suffer the same flaw.

4.3. Model Selection

We found that the GAN and VAE-GAN models did not improve monotonically with training, and using the final trained model would generally be far from optimal. Figure 4 shows a typical example of the variability of pixel-wise CRPS scores, and RALSD scores, for the GAN and VAE-GAN as a function of generator training samples. Each new checkpoint represents the generator seeing an additional 6,400 20×20 images. The training instability may have been alleviated with a more sophisticated treatment of the learning rate. Instead, we adopted the simple strategy of saving the generator model weights at 100 intermediate checkpoints during training, and evaluating the final one-third of these models on the 2019 validation data.

We selected a “best” model by looking at the evaluation results for checkpoints that produced the best pixel-wise CRPS scores, then looking manually at the full and thresholded rank histograms of these. A final, best model checkpoint was selected and then evaluated on the hold-out 2020 data set. The final models that we selected were the GAN generator saved after 460,800 training samples, and the VAE-GAN generator saved after 550,400 training samples.

5. Results

Unless stated otherwise, all quantitative results in this section are produced using 256 randomly-chosen examples from the unseen 2020 data set. The same examples are used for all models, and for all the different metrics assessed. For all stochastic models, including our own neural networks, we draw 100 ensemble members for each example.

5.1. Description of Alternative Methods

We compare our approach to a number of different methods, which we describe here briefly. These include very simple methods, such as naïve upsampling and Lanczos filtering (Turkowsky, 1990), intermediate methods such as Rainfall Filtered Auto Regressive Modeling (RainFARM) (Rebora et al., 2006), and sophisticated methods such as the ecPoint approach (Hewson & Pillosu, 2021), and a deterministic neural network trained on mean squared error.

ecPoint (Hewson & Pillosu, 2021) is an ECMWF statistical post-processing technique that gives a probabilistic prediction for rainfall intensity at a specific point, accounting for sub-grid variability and model biases. This is done by assigning the parent grid-cell into one of over 100 bins (categories), based on the atmospheric conditions predicted by the model. Mapping functions, which scale the precipitation multiplicatively, are pre-calculated for each bin based upon the percentiles of the training data. The input fields for the ecPoint approach are similar to, but not exactly the same as, the input fields to our generative model. The ecPoint model was originally trained with global station observations, using precipitation accumulated over 12 hr. For our application, we use the same atmospheric variable decision tree “break-points” as the standard ecPoint implementation, but naïvely converted to work with hourly data—accumulated quantities are divided by 12, other quantities are left unchanged. We then retrain the probability mapping functions on our output data set: NIMROD, at hourly intervals. However we appreciate that this implementation of the ecPoint approach is somewhat flawed, and ideally optimal break-points would be re-derived for hourly data.

While ecPoint was designed to sample gridbox uncertainty, the method was first designed as a post-processing tool and as such only describes how to get a point-wise probabilistic prediction of the possible sub-grid values within an IFS gridbox. To use it as a downscaling tool requires a choice on how to sample multiple high-resolution grid points within the same IFS gridbox. We use two example approaches that maintain the correct pixel-wise statistics in the high-resolution output. *ecPoint no-corr* refers to a “no-correlation” method of sampling every pixel independently from the parent distribution. Visually, this would lead to a very noisy image. Quantitatively, accumulated rainfall forecasts over larger regions will have insufficient variation, since there is no spatial coherence to the forecast. *ecPoint part-corr* refers to a “part-correlation” method in which the same sampled value is used for every pixel within one low-resolution parent grid cell. This is equivalent to the standard use of ecPoint, where the input image is post-processed but not upsampled. Visually, this would produce a blocky image, but accumulated rainfall over larger regions may vary more realistically. We always use 100 ensemble members, which allows us to *permute* the 100 candidate ecPoint predictions at each pixel, that is, sample without replacement. This improves the CRPS very slightly compared to sampling *with* replacement. Again, we emphasize that these steps are not part of the core ecPoint approach, but are merely simple methods for combining single-pixel probabilistic predictions into complete images.

RainFARM is a downscaling method which has been developed specifically for rainfall. It is based on non-linearly filtering the output of a linear auto-regressive process, whose properties are derived from the information available at the large scales. This process extrapolates the large-scale spatio-temporal power spectrum of the meteorological predictions to the small, unresolved scales. The basic concept is to preserve the amplitude and phases of the original field at the scales with high confidence in original model prediction, and to reconstruct the Fourier spectrum at the smaller (unreliable, unresolved) scales. RainFARM can be used stochastically to generate multiple predictions for the same input.

Lanczos filtering is a traditional, interpolation-based widely used image scaling method. It is perhaps better than “constant upsampling” of the input by repetition, although in our application the two approaches are very similar. We also use a deterministic convolutional neural network method that has the same architecture as the GAN generator, with the noise input removed, and is trained on a mean squared error loss function. Lanczos filtering, constant upsampling, and this CNN are all deterministic.

5.2. Model Evaluation

Table 1 shows numerical results for the pixel-wise and pooled CRPS scores, the RALSD score, and RMSE, for the GAN and VAE-GAN models we developed, compared to existing models/approaches. For each of the CRPS metrics, the best score is obtained by the VAE-GAN model, marginally ahead of the GAN. Notably, the GAN and VAE-GAN compare favorably to ecPoint on pixel-wise CRPS, despite the latter being a very well-calibrated

Table 1

Table Showing Evaluation Results for Different Models, on Previously Unseen 2020 Data, for CRPS, Power Spectra Error (RALSD), and RMSE

Model	Evaluation metric							
	CRPS (mm/hr)					RALSD (dB)	RMSE (mm/hr)	
	Pixelwise ^a	Avg 4 ^a	Max 4 ^a	Avg 16 ^a	Max 16 ^a		Ens-mean	Individual
GAN	0.0856	0.0844	0.1151	0.0806	0.2117	4.88	0.404	0.528
VAE-GAN	0.0852	0.0840	0.1147	0.0802	0.2104	5.34	0.405	0.499
ecPoint no-corr ^b	0.0895	0.1075	0.3987	0.1195	1.5948	16.35	0.423	0.644
ecPoint part-corr ^b	0.0895	0.0889	0.1255	0.0883	0.2485	9.78	0.423	0.644
RainFARM	0.1331	0.1332	0.1697	0.1286	0.2888	9.95	0.442	0.444
Lanczos ^c	0.1412	0.1392	0.1731	0.1309	0.2923	15.38	0.447	
Det CNN ^c	0.1347	0.1325	0.1644	0.1250	0.2817	16.74	0.404	

Note. The best score for each metric is highlighted in bold.

^aThese correspond to different methods of spatial pooling, as described in Section 4.2.1. ^bThe two ecPoint variants have identical pixel-wise statistics, by construction. ^cThese are deterministic methods, hence the CRPS reduces to the mean absolute error, and there is no separate ensemble-mean RMSE.

point-wise method. We consider this a strong result. The deterministic model scores poorly, showing the added value of the stochastic nature of the generative approaches. The RALSD scores show that only the GAN and VAE-GAN produce images with realistic power spectra. The RALSD figures for both ecPoint variations are included, but ecPoint is not designed to produce coherent spatial forecasts so these scores are unimportant. Ensemble-mean RMSE values are given too: the GAN produces the best score here, marginally better than a deterministic CNN trained to minimize mean squared error. The VAE-GAN ensemble-mean RMSE is only slightly worse, and these all score notably better than the other methods. Individual prediction RMSE values are given for completeness, but, as discussed in Section 4.2.3, it is a poor metric to optimize for due to the double penalty effect.

Figure 5 shows plots for four example cases. More detailed descriptions of these meteorological scenarios can be found in Supporting Information S1. The examples give a clear indication of how our GAN model produces more detailed and more visually realistic images than any other method, as well as being more robust at forecasting more intense rainfall. The RainFARM algorithm does produce some small-scale detail compared to the IFS input, but it is limited to producing the same texture everywhere in the image and it does not reproduce the overall structure of the high-resolution truth as well as the GAN, nor does it predict extremes of rainfall missed by the IFS forecast. The ecPoint mean prediction is shown, for completeness, and is effectively a bias-corrected IFS. However, for clarity, none of our quantitative methods use this ecPoint mean; instead they use ecPoint ensemble members constructed via the *no-corr* and *part-corr* methods described previously. Finally, the deterministic CNN trained on mean squared error produces very “blurry” predictions. This model often greatly over-predicts the spatial extent of very light rainfall, and is incapable of predicting extremes. In general, the smoother plots with less variance and fine-scale structure are rewarded by the RMSE metric, but punished by the RALSD metric (further details in Section 4.2.3, with results displayed in Table 1).

A set of example predictions for the best GAN and VAE-GAN models are shown in Figure 6. For each example, Figure 6 shows three different ensemble predictions from each model. The same randomly selected cases are shown in this example, encompassing a range of precipitation conditions. The predictions produced by these models provide a high-quality solution set to a range of different meteorological conditions. There is sharply varying spatial structure in the predictions that is reminiscent of the true conditions, and not produced by any of the existing approaches.

We can clearly see from these examples that the GAN and VAE-GAN models are very capable of improving on the IFS forecast and bringing the predictions closer to the truth. Further, both models produce multiple realizations for the same situation, giving a clearer idea of the uncertainty. The main improvement offered by the GAN over the VAE-GAN is an improved tendency to predict more intense rainfall.

Example predictions for different input conditions

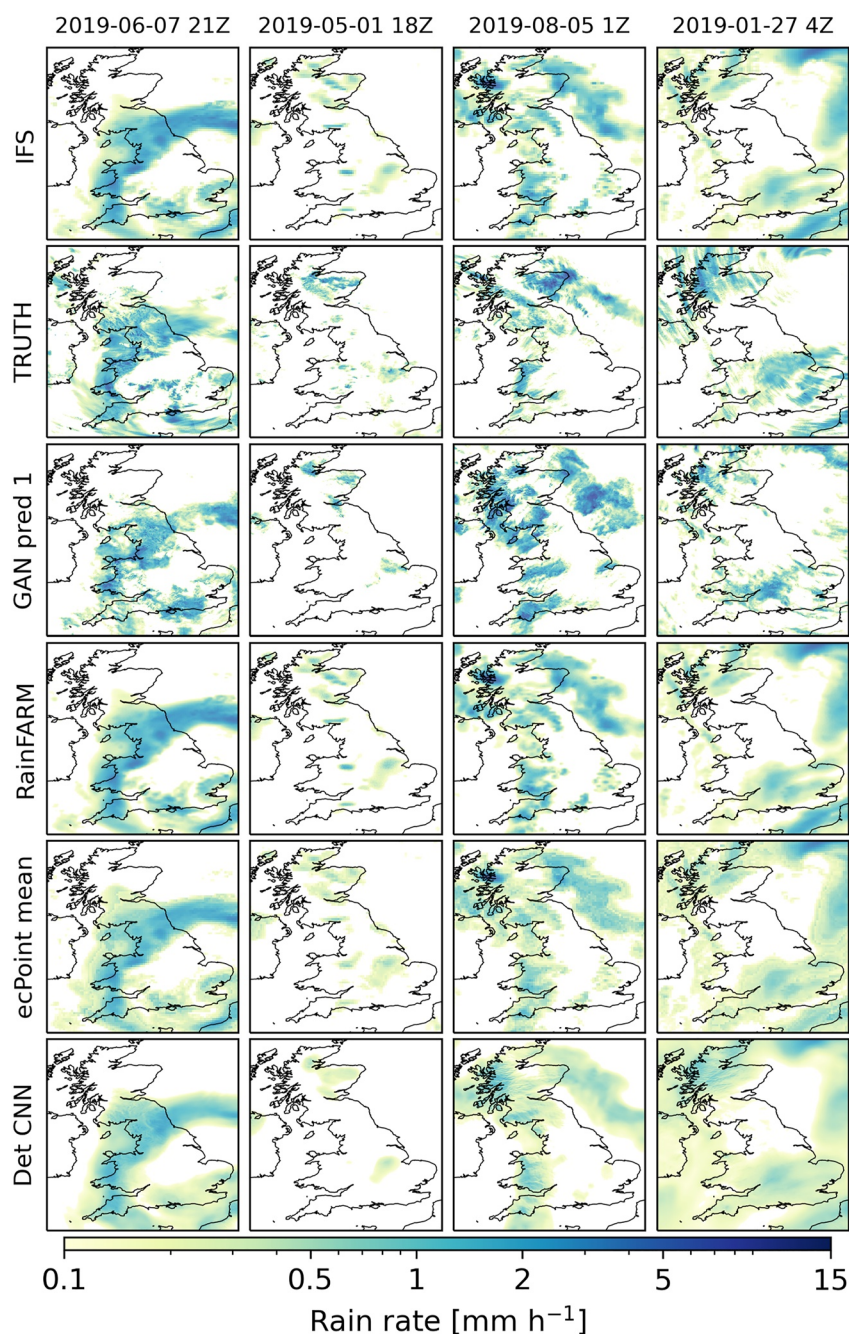


Figure 5. Comparison of predictions generated by the generative adversarial network (GAN) with those produced by existing methods, for four randomly-chosen cases. The following fields are shown: the Integrated Forecast System (IFS) forecast low-resolution input data, the NIMROD high-resolution ground truth data, a single GAN ensemble member prediction, a RainFARM prediction, the mean ecPoint prediction, and a deterministic neural network applied to the IFS data.

5.3. Model Predictions for Extreme Events

Since machine-learned models are trained on historic data, they often struggle with extreme events. However, these are some of the most important situations to forecast accurately and reliably, so we were particularly

Example predictions for different input conditions

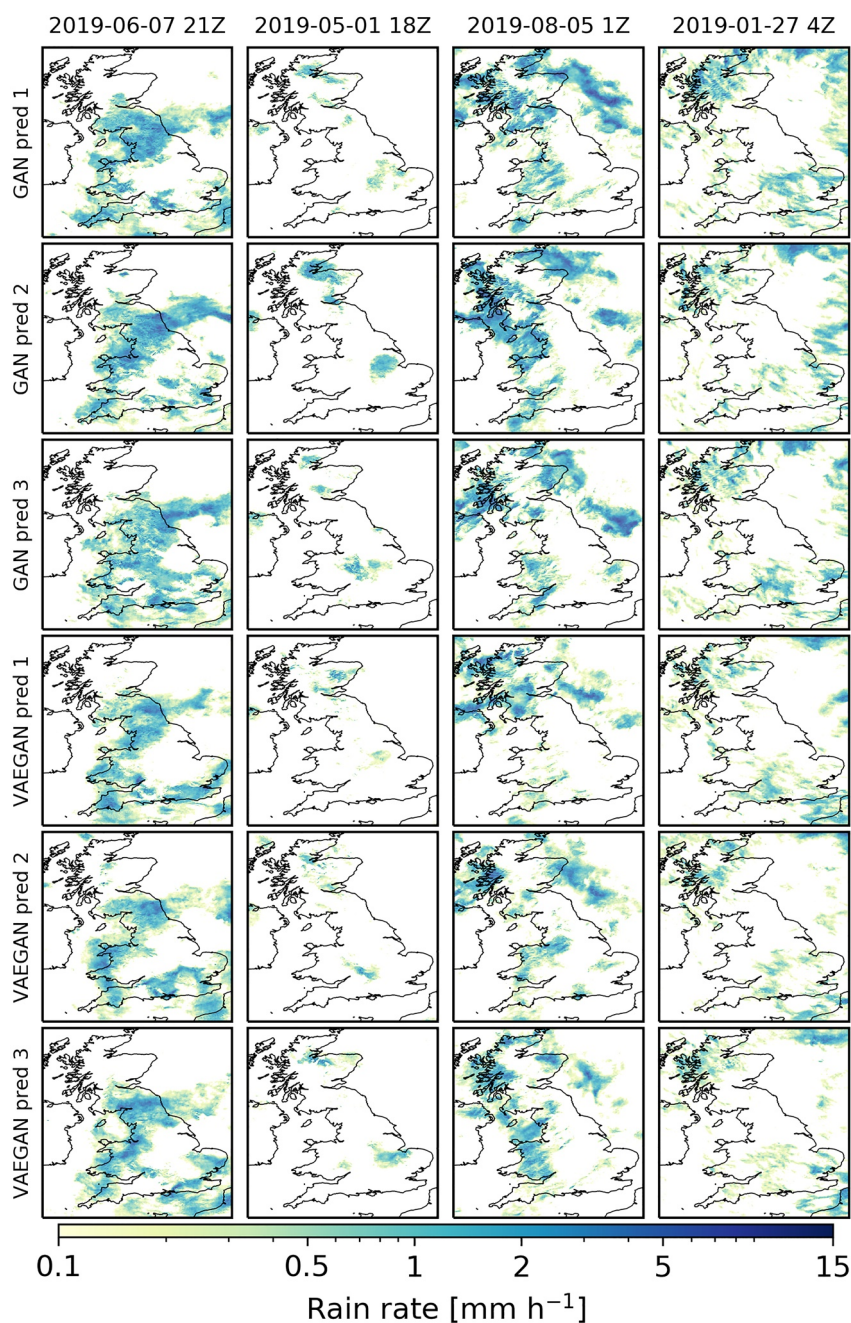


Figure 6. Examples of multiple generative adversarial network (GAN) and variational auto-encoder-generative adversarial network (VAE-GAN) ensemble predictions for four different input examples. The examples used are the same as in Figure 5.

interested in assessing our models' performances on extreme events.

Figure 7 shows the GAN and VAE-GAN model responses to one of the most extreme rainfall events in our data set. These data points are taken from 09:00–10:00 UTC of the 9 February 2020, during which there was a significant rainstorm across the UK, named Storm Ciara. Figure 7 shows input fields including total precipitation, convective precipitation, orography and a wind quiver plot, as well as the truth data, a single example prediction

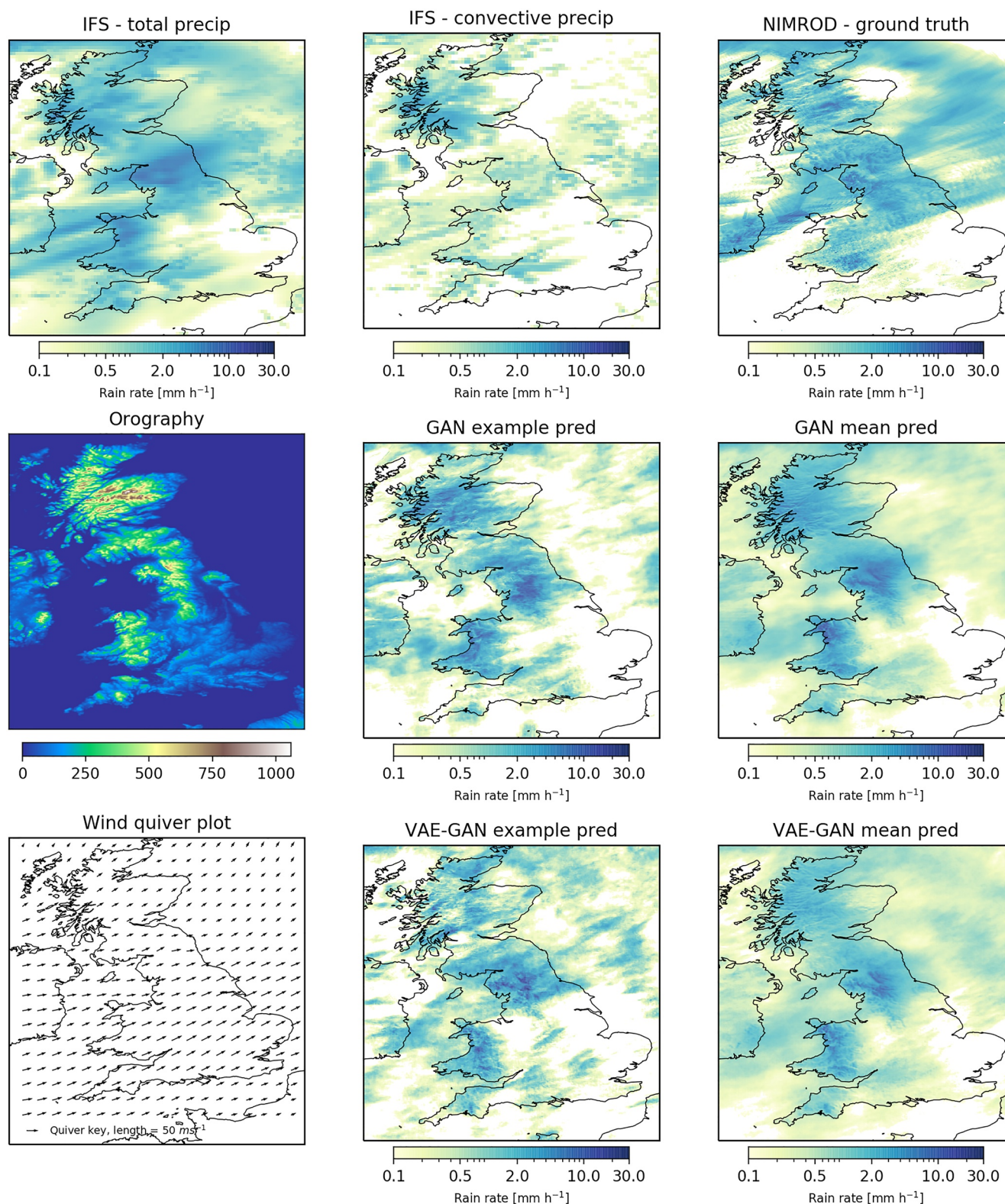


Figure 7. Generative adversarial network (GAN) and variational auto-encoder-generative adversarial network (VAE-GAN) model output example predictions and means for one of the most extreme examples in our data set, from 09:00-10:00 UTC of the 9 February 2020, showing the total precipitation and wind direction and strength from the IFS forecast, orography, and the ground truth NIMROD data. Note that the colorbar has been changed from the previous set of examples and now ranges from 0.1 to 30 mm.

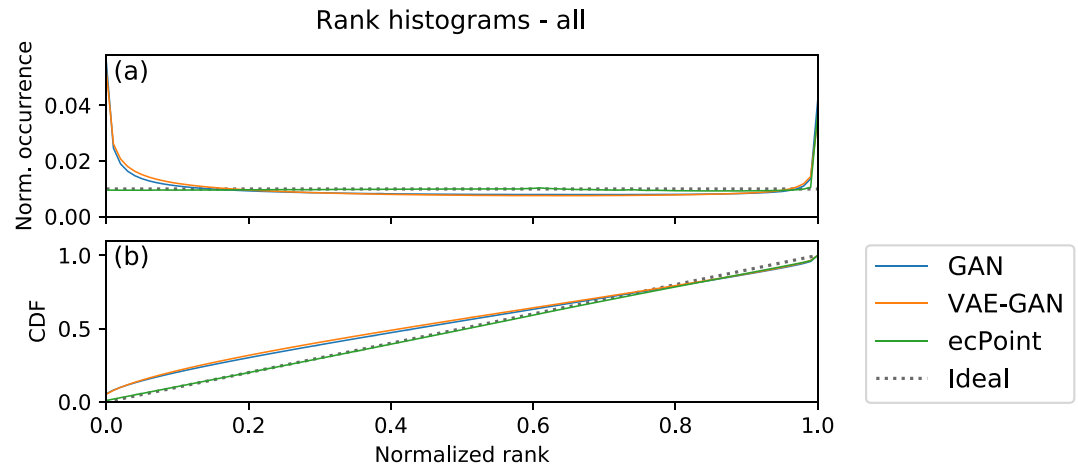


Figure 8. Calibration plot across all events: (a) shows the frequency of per-pixel normalized ranks for the trained generative adversarial network (GAN) and variational auto-encoder-generative adversarial network (VAE-GAN) models evaluated on the hold-out data set (2020), compared to the ecPoint approach. The dotted gray line shows the ideal distribution for comparison. (b) Shows the same as panel (a), except displaying the cumulative distribution functions (CDFs) of the distributions.

and the mean prediction. The GAN and VAE-GAN models capture the peak intensities and fine-scale structure of the rainfall event better than the IFS forecast.

5.4. Rank Statistics

Figure 8 shows the pixel-wise rank distributions of the GAN and VAE-GAN ensembles, based on 100-member ensembles. These plots show that the majority of the outlier ranks are in the tail ends of the distribution, where r is either close to 0 or 1, implying that the GAN and VAE-GAN are slightly underdispersive. The GAN marginally out-performs the VAE-GAN on this metric. The ecPoint approach outperforms our networks on this metric considerably; however, ecPoint is essentially optimized for this metric, as its *raison d'être* is to produce well-calibrated pointwise forecasts. Our approaches, on the other hand, also try to produce realistic larger-scale spatial structures. The ecPoint approach is still somewhat underdispersive on the right-hand tail, though.

Figure 9 shows the same analysis but now restricted to “extreme events” – the top 0.01% of forecasted precipitation events seen in the IFS input. The GAN now outperforms the VAE-GAN, but both are under-dispersive, particularly on the right-hand tail. The ecPoint approach is now over-dispersive, with the real sample rarely falling in the bottom or top 20% of predictions. This is perhaps related to the multiplicative ansatz of the ecPoint

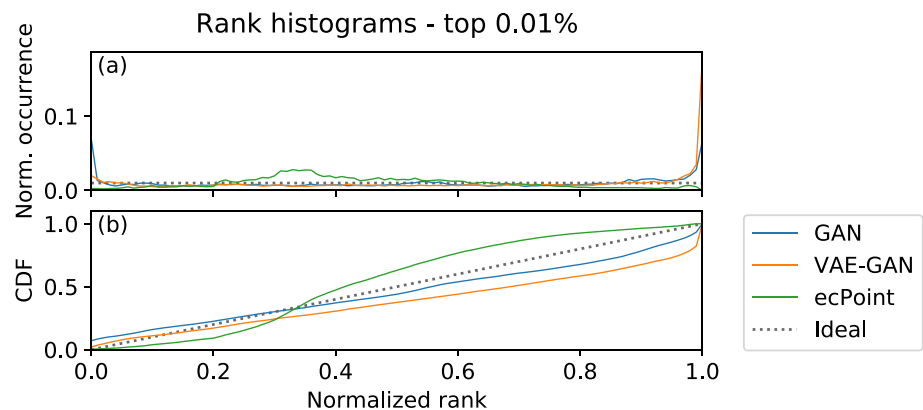


Figure 9. Calibration plot; similar to Figure 8, but only for the top 0.01% of Integrated Forecast System (IFS) forecasted precipitation events. This corresponds to IFS predictions above 5.7 mm/hr of precipitation; 226 of these are present in the 256 94×94 input images used.

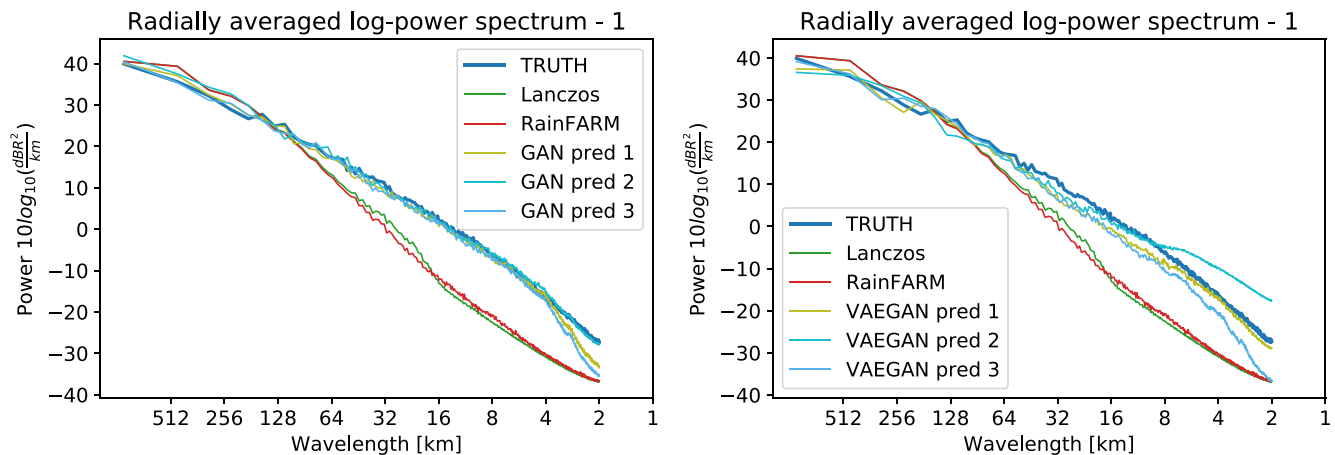


Figure 10. Plot displaying the radially-averaged power spectrum of images with decreasing scale produced by the generative adversarial network (GAN) and variational auto-encoder-generative adversarial network (VAE-GAN) models, on the first example situation, compared to both Lanczos interpolation and the RainFARM method.

approach. Among the three methods, the GAN perhaps performs best on extreme events, but none of the three methods are particularly reliable on these.

5.5. Power Spectra

In addition to the scores displayed in Table 1 for the RALSD, Figure 10 shows radially averaged power spectral density (RAPSD) plots for the GAN and VAE-GAN models for the first of the four example situations, compared to the ground-truth NIMROD data, and the existing RainFARM and Lanczos models. Details of the RAPSD implementation are given in Section 4.2.3. These plots show, first, that the RainFARM and Lanczos models are missing a lot of detail at lower scales (really, any grid scale under ~ 100 km), which is unsurprising as this is identifiable by eye in all of the example plots. The GAN and VAE-GAN are both much closer to the truth in terms of retaining energy in the image at much finer scales. Only one example is analyzed here (example 1 in Figure 5); further examples are included in Supporting Information S1. Interestingly, in this particular example, the VAE-GAN contains more information than the GAN at the smallest scales. This is not always the case, as shown in Table 1. There is also typically more variation between members of the GAN ensemble than those of the VAE-GAN, unlike in this particular case.

5.6. ROC Curves

The ROC curves for the GAN, VAE-GAN and ecPoint (partial-correlation and no-correlation) models are shown here for the 0.5 and 5 mm/hr thresholds, using pixel-wise analysis. Additional plots using spatial pooling, and for other precipitation thresholds, are included in Supporting Information S1. A perfect prediction would yield a point in the upper left corner of the ROC space, representing 100% sensitivity (no false negatives) and 100% specificity (no false positives). The dashed diagonal line represents random chance. Consequently, points far above the diagonal represent good classification results.

For the 0.5 mm/hr threshold, shown in Figure 11, the GAN and VAE-GAN slightly outperform the ecPoint approach. The GAN and VAE-GAN lines are generally above and left of the ecPoint line, and have the largest areas under the curve. For the 5 mm/hr threshold, shown in Figure 11, the results are harder to interpret. The curves are somewhat distorted due to the finite ensemble size and the rarity of the event (the event frequency is 0.001). Although the ecPoint line has the largest area under the curve, the GAN line is the furthest left in the initial portion of the graph, followed by the VAE-GAN. This likely equates to better performance in the limiting case of unlimited ensemble members (see Ben Bouallegue and Richardson (2022) for more on interpreting the area under ROC curves in the case of rare events).

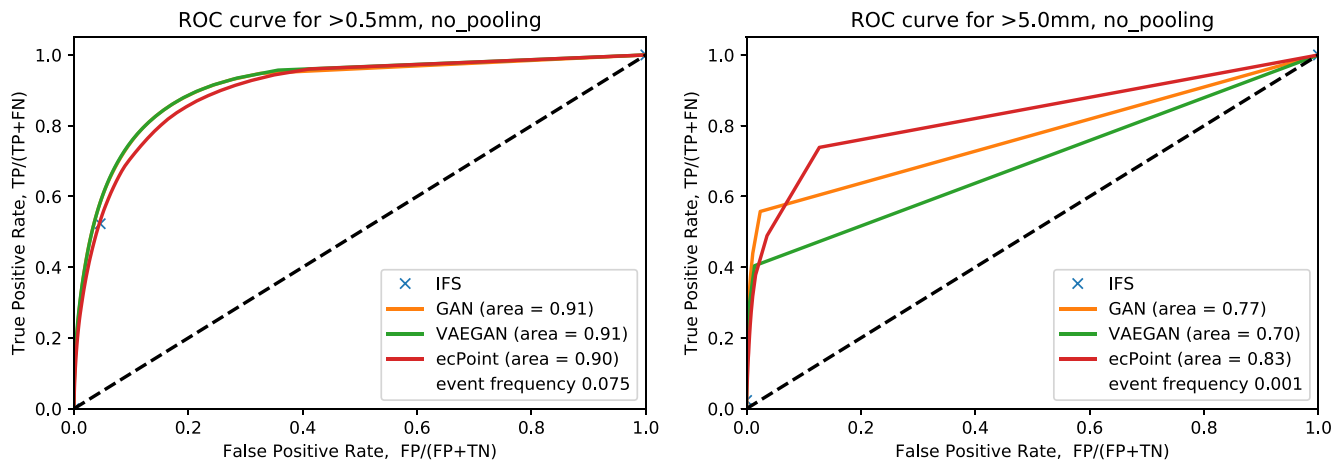


Figure 11. ROC curves for the generative adversarial network (GAN), variational auto-encoder-generative adversarial network (VAE-GAN) and ecPoint models for 0.5 and 5.0 mm/hr precipitation thresholds. The upsampled input data, labeled “IFS”, is represented by a cross, as it is a single prediction rather than an ensemble.

5.7. Fractions Skill Score

Fractions skill score (FSS) curves for the GAN and VAE-GAN models are plotted in Figure 12. FSS curves are shown here for the 0.5 and 5 mm/hr thresholds only, with additional plots for 0.1 and 2 mm/hr in Supporting Information S1.

For the light (0.5 mm/hr) rain threshold, both the GAN and VAE-GAN models produce a noticeably better ensemble FSS than the ecPoint variants, and have useful skill even at the pixel level. The FSS of individual *ecPoint no-corr* members is particularly high at intermediate spatial scales, but we believe this is an artifact of the metric when applied to very “noisy” images, as discussed in Section 4.2.5, and not a sign of genuinely useful output. The individual GAN members have a higher FSS than the individual VAE-GAN members, and the GAN ensemble FSS is better than that of the VAE-GAN.

For the heavy (5.0 mm/hr) rain threshold, the GAN significantly outperforms the VAE-GAN for both ensemble and individual member FSS. The VAE-GAN struggles at producing the highest intensities of precipitation. The GAN ensemble clearly outperforms the ecPoint ensemble at small and intermediate spatial scales. At the largest

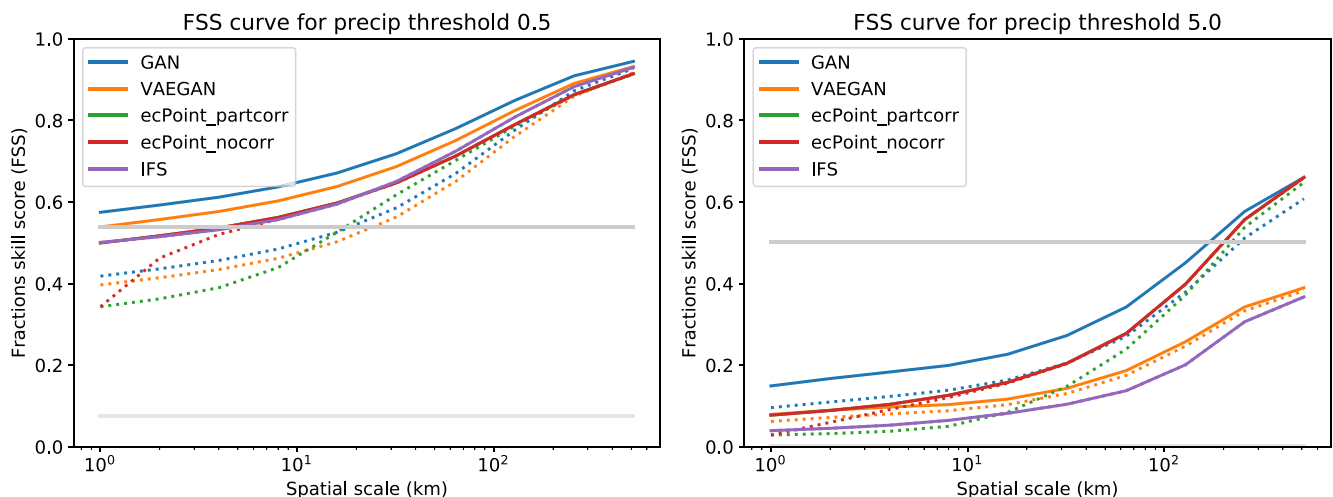


Figure 12. Fractions skill score (FSS) curves for the generative adversarial network (GAN), variational auto-encoder-generative adversarial network (VAE-GAN) and ecPoint models. The solid lines represent “ensemble FSS scores”, as described in Section 4.2.5, while the dashed lines represent basic FSS scores applied to individual ensemble members. The gray lines represent commonly-used “no-skill” and “useful skill” thresholds, of p and $\frac{1+p}{2}$, where p is the event probability.

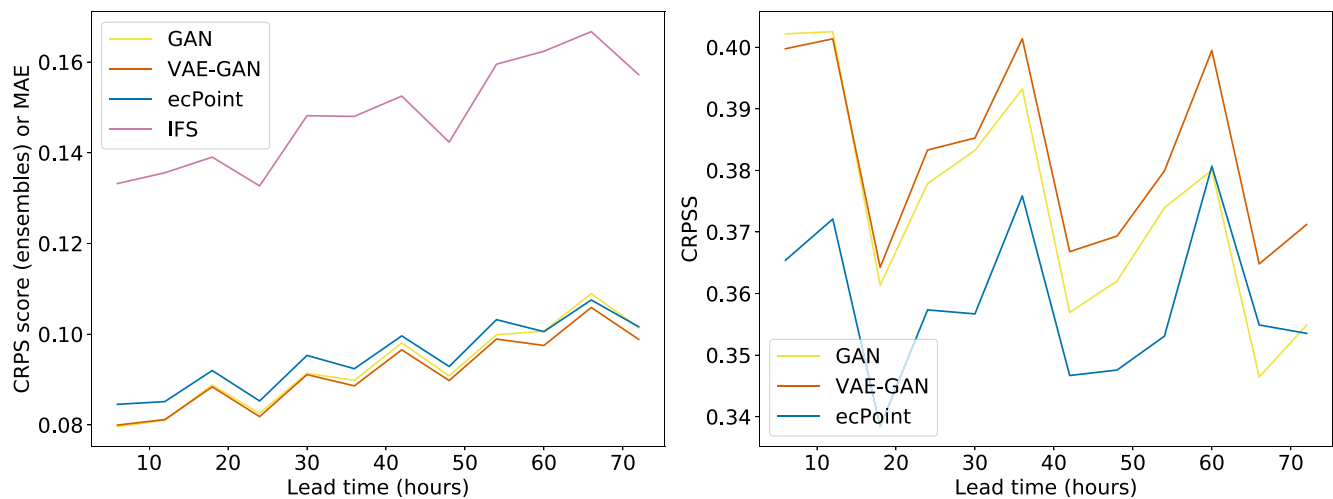


Figure 13. Scores for the generative adversarial network (GAN), variational auto-encoder-generative adversarial network (VAE-GAN) and ecPoint models with increasing lead time, compared to the baseline case of the IFS forecast.

spatial scales, the methods perform similarly, reflecting similar skill at predicting the overall extreme event frequency.

5.8. Model Performance With Increasing Lead Time

To minimize the difference between our model and truth data sets during training we restricted training to lead times between 7 and 17 hr. Previous results were assessed solely on this time period. However, we are also interested in using our tool on shorter and particularly on longer lead times. Figure 13 shows plots of the pixel-wise CRPS and CRPSS ($=1 - \frac{\text{CRPS}}{\text{CRPS}_{\text{IFS}}}$) metrics for the GAN and VAE-GAN models, for increasing lead time, compared to the IFS forecast data and the ecPoint approach. These are obtained *without* retraining the model with data from other lead times.

The lead time investigation was carried out using every available 00Z forecast in our 2020 data set, at lead times every 6th hour from 6 to 72 hr, again with an ensemble size of 100. All models show generally increasing CRPS with lead time, with visible diurnal cycle effects. The GAN and VAE-GAN show decreasing CRPSS as lead time increases, consistent with them being applied on longer lead times than they were trained on. However, the ecPoint approach shows CRPSS increasing with lead time, despite also being calibrated on 7–17 hr data. The GAN, VAE-GAN and ecPoint approach all have somewhat similar CRPS scores, compared with the IFS input data. The GAN and VAE-GAN slightly out-perform the ecPoint approach, although the GAN is overtaken by ecPoint at the longest lead times.

5.9. Pure Super-Resolution Tests

Due to the different origins of our input data (IFS) and truth data (NIMROD), we are asking any machine learning solution to undertake two tasks: super-resolution, and bias/spread correction to account for forecast error. As a sanity check of our models, we also trained them to instead ingest coarsened NIMROD data (area-averaged to 0.1°) and full-resolution geographic fields, and predict the full-resolution 0.01° NIMROD field. The resulting problem is close to that tackled in Leinonen et al. (2020), without the temporal component, and is inherently easier than our full problem. The value of these experiments is to establish the limits of our ML models, and to understand whether the performance is limited by the super-resolution component or the forecast error correction component of the problem. The resulting models from this experiment are unlikely to perform well on the full downscaling problem since coarsened NIMROD radar data is not interchangeable with IFS forecast data. We have included results from this study in Appendix B.

6. Discussion and Conclusions

We present two models: GAN-based and VAE-GAN-based, both capable of increasing the resolution of forecast data by a factor of 10 and calibrating the forecast. Both models demonstrably add skill to the forecast, and produce similar forecasts with similar spatial structure. The VAE-GAN produced slightly better CRPS scores than the GAN. However, the GAN is better-calibrated than the VAE-GAN, is more capable of producing intense rainfall than the VAE-GAN, and perhaps produces slightly more large-scale variation than the VAE-GAN. Both models produce much better results than simple alternative methods, and achieve similar or slightly improved scores compared to the precipitation downscaling state-of-the-art ecPoint method, whilst producing spatially coherent and visually realistic images, which are easier to interpret. Both the GAN and VAE-GAN models allow for an ensemble of predictions to be produced, providing an estimate of uncertainty quantification, which is essential in weather forecasting applications. From the perspective of the rank histograms, Figure 8, ecPoint produces a better calibrated ensemble than either GAN method. This is consistent with the design of ecPoint which was configured to produce a well-calibrated ensemble. It is interesting that both GANs approach the calibration of ecPoint, in this plot and the ROC plots, with no explicit training for calibration. Furthermore, when restricted to extreme events, the neural network approaches are roughly as miscalibrated as ecPoint.

To better understand the bounds of our success we also used the same GAN-based architecture to carry out a NIMROD to NIMROD mapping, similar to that of Leinonen et al. (2020), where the input field was an average-coarsened version of the NIMROD precipitation field (plus the static fields). For this changed problem we see a significant drop in CRPS (from 0.0856 to 0.0230 mm/hr) and improvement in calibration. We interpret these auxiliary results, which are presented in the appendix, as a demonstration that the difference between the IFS and NIMROD data sets in the full problem (i.e., *forecast error*) is the main factor limiting the success of the model. This is caused by the misalignment of fronts and other precipitation events between the two data sets. Future work could explore methods to limit these effects, for example, pre-processing the data set to include only well-aligned events.

We believe there are several avenues for future exploration with this work. Foremost is the application of our model to the downstream tasks of flood modeling, where the bias correction and higher resolution could help improve the accuracy of the flood forecast. We would like to investigate the potential of applying the model to post-process *ensembles* of forecasts, which may guide the network further toward an accurate assessment of forecast error. Price and Rasp (2022) incorporated this data in their approach, but did so by fixing the number of ensemble members ingested. Using a transformer architecture in the ensemble dimension could be an exciting, ensemble-size-agnostic, approach to try. There may be some advantages to a split approach, in which coarse-scale bias/spread correction is applied before the downscaling step. Further work with more significant computational power could re-visit the temporal aspect, to give temporally consistent downscaled results. We barely explored modifications to the network architecture, and gains could perhaps be made here. We expect our model could be extended to work more generally across different geographical regions. Publicly available data sets could be used to build a downscaling model that could be applied globally, although this would have challenges in areas of the world where reliable truth data is not presently available. More generally, further work will be required to produce an operational-ready product.

Appendix A: Ablation Studies

A1. Varying Training Data Distribution

As discussed in Section 2.4, the training data was pre-processed into sub-images and sorted into bins according to the proportion of pixels containing rain within the sub-image. We then trained the network on different frequency distributions of these bins. We anticipated that the model would otherwise under-predict precipitation if trained on images that overwhelmingly do not contain precipitation. We treated this training data distribution as a hyper-parameter to be optimized, and explored a few different distributions.

The initial selection of sampling equally from these bins caused the GAN to over-predict rainfall, shown in Figure A1. We also investigated training the network on the natural distribution of images—sampling in the proportion that the data would naturally fall. This corresponds to 41× as many images from the least-rainy bin as the most-rainy bin. Sampling the sub-images in this way produced marginally better CRPS scores than the

Example predictions for different input conditions

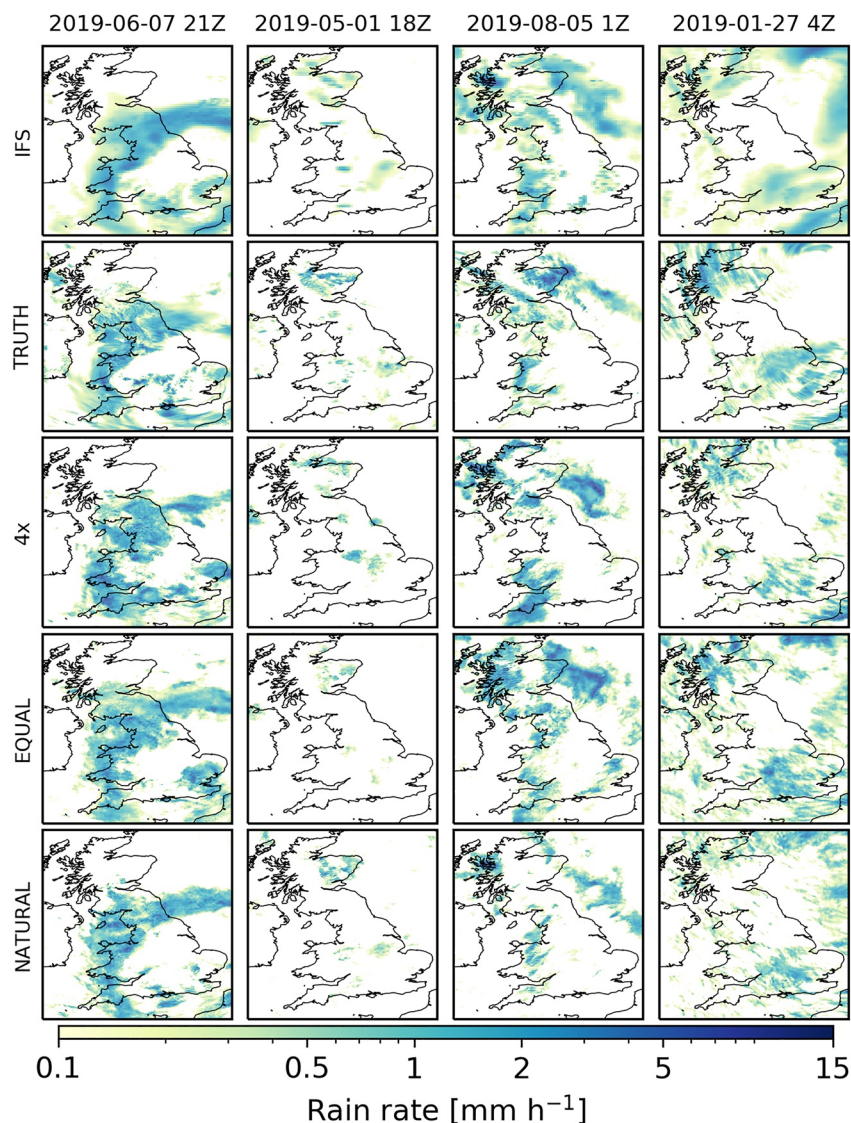


Figure A1. Example generative adversarial network (GAN) model output predictions with different input sample weighting, compared to the input low-res information (IFS) and hi-res ground-truth data (NIMROD).

other distributions trialled, except for the main GAN (see Table A1). However, as anticipated, the network clearly under-predicts rainfall.

Finally, we tried showing the network k times as many images from the “least-rainy” bin as the “most-rainy” bin, with k varying between 2 and 12. The intermediate bin weights were interpolated linearly between these. Lower k -values tended to produce better rank histogram plots and RALSD scores, and higher k -values produced better results for CRPS. A k -value of 4 was determined to offer the best results overall. Example predictions for our best model are shown in Figure A1. Although the network still has a tendency to over-predict light rainfall, it retains the predictive power at the extremes of rainfall, so this was determined to be the best compromise.

The plots and evaluation numbers shown here are for the final, best version of the model. The differences here are subtle. The initial assessment of the data training weights was carried out on a more preliminary version, and

Table A1
Table Showing Evaluation Results for Our Final GAN, and Various Ablated Versions

Variant	Evaluation metric							
	CRPS (mm/hr)					RALSD (dB)	RMSE (mm/hr)	
	Pixelwise	Avg 4	Max 4	Avg 16	Max 16		Ens-mean	Individual
Main GAN	0.0856	0.0844	0.1151	0.0806	0.2117	4.88	0.404	0.528
“Natural” ^a	0.0877	0.0866	0.1185	0.0832	0.2192	5.10	0.416	0.514
“Equal” ^a	0.0912	0.0903	0.1226	0.0871	0.2272	4.33	0.417	0.533
No CL ^b	0.0901	0.0890	0.1215	0.0857	0.2247	5.64	0.419	0.502
No geog ^c	0.0857	0.0845	0.1159	0.0805	0.2155	4.43	0.407	0.541 ^a

Note. The best score for each metric is highlighted in bold.

^aVarying the distribution of data used during training. ^bNo content loss term used during generator training, only discriminator loss. ^cNo geographic fields used, that is, the high-resolution orography and land-sea mask.

the differences were much more stark. Improvements on the 4x-trained model also significantly improved models trained with other input data distributions, however, we still consider the balance of training data an important factor to be considered. Choosing an optimal input data distribution should also help to accelerate training, even if the respective models eventually converge to similar minima.

A2. Removal of Content Loss Term

In Section 4.1, we described that the generator is not just trained on discriminator loss, but an ensemble-mean-MSE content loss term is also added to the loss function. As shown in Table A1, this content loss term improves the resulting network considerably. The CRPS improves noticeably, and the RALSD and ensemble-mean RMSE also improve.

A3. Removal of Geographic Fields

Our main models used not just low-resolution IFS forecast fields as input, but also high-resolution orographic and land-sea mask fields, since these are expected to affect precipitation locally due to physical principles. From Table A1, it would appear that a network without these high-resolution inputs has roughly equivalent skill. However, we found that removing geographic fields made the network perform noticeably worse on the 2019 validation data set when averaged across a larger number of candidate model checkpoints, and earlier versions of the model showed a reduction in skill from removing geographic fields. We therefore believe that the geographic fields should still be included, even if the improvement in skill was not apparent in the final model training runs used in this paper.

Appendix B: Pure Super-Resolution Problem

We also assessed the performance of the model architecture on the pure super-resolution problem of increasing spatial resolution without accounting for forecast error. The input data is now NIMROD radar data that has been coarsened, using averaging, by a factor of 10. The output is compared with the original, high-resolution NIMROD truth. We again pass in high-resolution orography and land-sea masks to the model. The precise information flow and network architecture is detailed in Figure 2, with the “9 IFS fields” input replaced by a single “coarsened NIMROD” input.

Example predictions for a model trained to map from coarsened to full-resolution NIMROD data are shown in Figure B1. It is immediately clear that the model performs very well on this task. Comparing these results to Figures 5 and 6 indicates how much more challenging the full downscaling problem is, in the presence of forecast error. Quantitatively, the model obtains a CRPS of 0.0230 mm/hr on this problem, less than one-third of the 0.0856 mm/hr CRPS obtained on the full problem. Increasing the resolution of the forecast appears to be a less

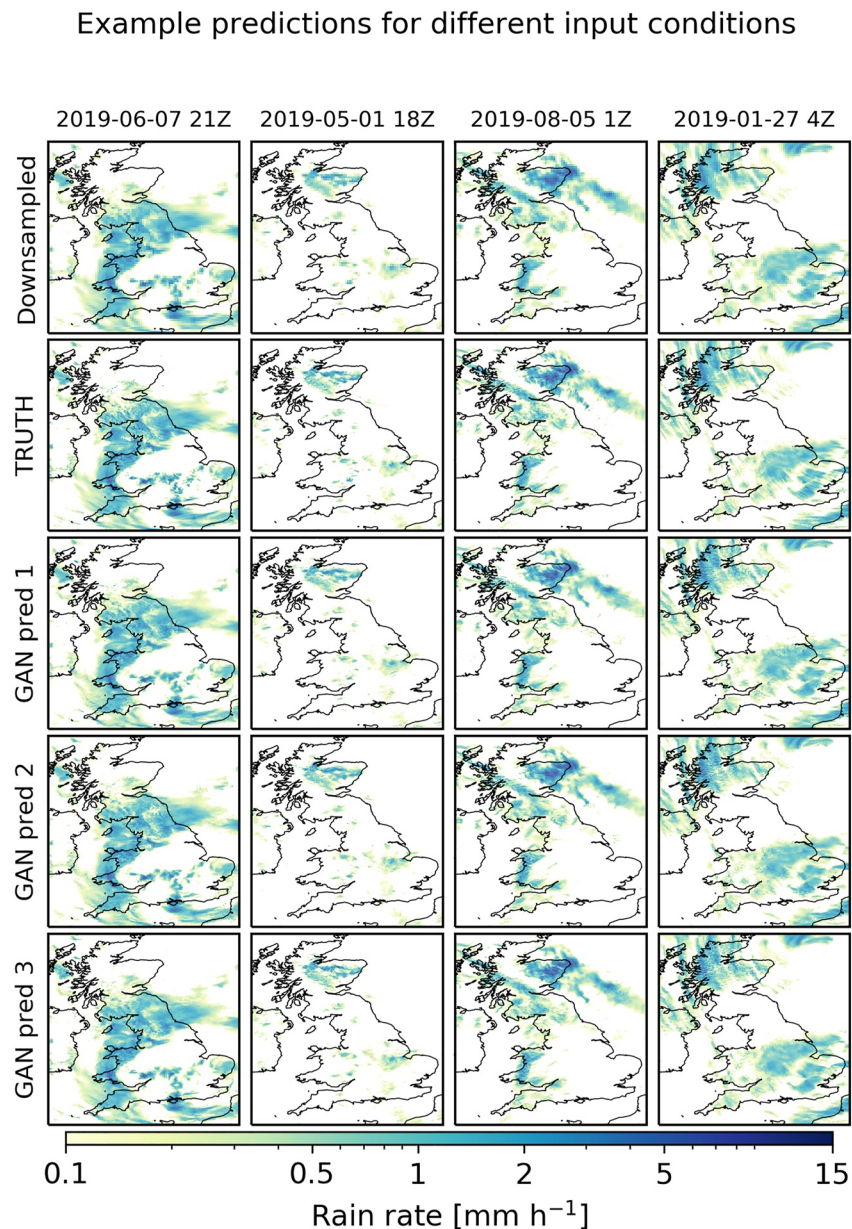


Figure B1. Generative adversarial network (GAN) model output predictions for the pure super-resolution problem: mapping coarsened NIMROD data to full-resolution NIMROD data.

challenging problem than accounting for forecast error in a spatially-coherent manner, and there is potential for an approach that performs these two steps separately.

Appendix C: Reduced Numerical Precision

The models were originally trained with 32-bit floating point numbers on an NVIDIA V100. We later gained access to an NVIDIA A100, on which TensorFlow automatically employed the “TensorFloat-32” (TF-32) format for many internal calculations. This number format has the range of 32-bit numbers but the precision of 16-bit numbers. The resulting trained models appeared to be equal in quality to the original models trained in 32-bit, although it is hard to be completely sure due to the random variations between different training runs.

We also tried using the TF-32 format only for inference, with models trained at full 32-bit. This was completely successful, giving practically identical metrics to 32-bit inference with approximately a 1/3 reduction in run time. We further tried training the model explicitly using 16-bit numbers. However, this quickly led to overflow during training and was unsuccessful.

Data Availability Statement

The code for the GAN and VAE-GAN models used in this paper is available at <https://doi.org/10.5281/zenodo.6922291>. A cleaned-up version of the code, with the same core functionality, is available at <https://github.com/ljharris23/public-downscaling-cgan>. We would recommend this for people looking to build on our work. Our code was adapted from Jussi Leinonen's GAN model, available at <https://github.com/jleinenon/downscaling-rnn-gan>. All experiments in this paper were performed within TensorFlow 2.7.0, except the deterministic CNN and the “natural” and “equal” training data ablation studies, which were performed within our older Tensorflow 2.2.0 environment. We did not find any scientific difference between models produced using the different Tensorflow versions. The ECMWF forecast archive can be obtained through MARS; more details are available at <https://www.ecmwf.int/en/forecasts/access-forecasts/access-archive-datasets>. MARS accounts for academic use are available for free, subject to certain conditions; see <https://www.ecmwf.int/en/forecasts/accessing-forecasts/licences-available>. The NIMROD radar data set can be obtained through CEDA; more details are available at <https://catalogue.ceda.ac.uk/uuid/27dd6ffba67f667a18c62de5c3456350>. A CEDA Archive account is required in order to access this data.

Acknowledgments

We are grateful to Stephan Rasp, Jussi Leinonen, Suman Ravuri and his colleagues at DeepMind, Peter Watson, Tim Hewson, Zied Ben Boualleague, Campbell Watson, Hannah Christensen, Milan Klöwer, and Fenwick Cooper for many useful conversations and ideas. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant No 741112, ITHACA). Computing resources were provided by the European Weather Cloud, which is an ECMWF and EUMETSAT project. PD gratefully acknowledges funding from the Royal Society for his University Research Fellowship as well as the ESIWACE project funded under Horizon 2020 No. 823988. PD and MC gratefully acknowledge funding from the MAELSTROM EuroHPC-JU project (JU) under No 955513. The JU receives support from the European Union's Horizon research and innovation programme and United Kingdom, Germany, Italy, Luxembourg, Switzerland, and Norway.

References

- Adewoyin, R. A., Dueben, P., Watson, P., He, Y., & Dutta, R. (2021). TRU-NET: A deep learning approach to high resolution prediction of rainfall. *Machine Learning*, 110(8), 2035–2062. <https://doi.org/10.1007/s10994-021-06022-6>
- Agrawal, S., Barrington, L., Bromberg, C., Burge, J., Gazen, C., & Hickey, J. (2019). Machine learning for precipitation nowcasting from radar images. Retrieved from <https://arxiv.org/abs/1912.12132>
- Applequist, S., Gahrs, G. E., Pfeffer, R. L., & Niu, X.-F. (2002). Comparison of methodologies for probabilistic quantitative precipitation forecasting. *Weather and Forecasting*, 17(4), 783–799. [https://doi.org/10.1175/1520-0434\(2002\)017<0783:comfpq>2.0.co;2](https://doi.org/10.1175/1520-0434(2002)017<0783:comfpq>2.0.co;2)
- Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. Retrieved from <https://arxiv.org/abs/1701.07875>
- Ben Boualleague, Z., & Richardson, D. S. (2022). On the ROC area of ensemble forecasts for rare events. *Weather and forecasting*, to appear. Retrieved from <https://www.preprints.org/manuscript/202111.0535/v1>
- Berrocal, V. J., Raftery, A. E., & Gneiting, T. (2008). Probabilistic quantitative precipitation field forecasting using a two-stage spatial model. *Annals of Applied Statistics*, 2(4), 1170–1193. <https://doi.org/10.1214/08-AOAS203>
- Bihlo, A. (2020). A generative adversarial network approach to (ensemble) weather prediction. Retrieved from <https://arxiv.org/abs/2006.07718>
- D'Onofrio, D., Palazzi, E., von Hardenberg, J., Provenzale, A., & Calmanti, S. (2014). Stochastic rainfall downscaling of climate models. *Journal of Hydrometeorology*, 15(2), 830–843. <https://doi.org/10.1175/JHM-D-13-096.1>
- Dong, C., Loy, C. C., He, K., & Tang, X. (2015). Image super-resolution using deep convolutional networks. Retrieved from <https://arxiv.org/abs/1501.00092>
- Duc, L., Saito, K., & Seko, H. (2013). Spatial-temporal fractions verification for high-resolution ensemble forecasts. *Tellus A: Dynamic Meteorology and Oceanography*, 65(18171), 1–22. <https://doi.org/10.3402/tellusa.v65i0.18171>
- Espoholt, L., Agrawal, S., Sønderby, C., Kumar, M., Heek, J., Bromberg, C., et al. (2021). Skillful twelve hour precipitation forecasts using large context neural networks. Retrieved from <https://arxiv.org/abs/2111.07470>
- Feser, F., Rockel, B., von Storch, H., Winterfeldt, J., & Zahn, M. (2011). Regional climate models add value to global model data: A review and selected examples. *Bulletin of the American Meteorological Society*, 92(9), 1181–1192. <https://doi.org/10.1175/2011BAMS3061.1>
- Gascón, E., Hewson, T., & Haiden, T. (2018). Improving predictions of precipitation type at the surface: Description and verification of two new products from the ECMWF ensemble. *Weather and Forecasting*, 33(1), 89–108. <https://doi.org/10.1175/WAF-D-17-0114.1>
- Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477), 359–378. <https://doi.org/10.1198/016214506000001437>
- Gong, B., Langguth, M., Ji, Y., Mozaffari, A., Stadler, S., Mache, K., & Schultz, M. G. (2022). Temperature forecasting by deep learning methods. *Geoscientific Model Development Discussions*, 1–35. <https://doi.org/10.5194/gmd-2021-430>
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial networks. Retrieved from <https://arxiv.org/abs/1406.2661>
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of Wasserstein GANs. Retrieved from <https://arxiv.org/abs/1704.00028>
- Hamill, T. (2000). Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review*, 129(3), 550–560. [https://doi.org/10.1175/1520-0493\(2001\)129<0550:iorthfv>2.0.co;2](https://doi.org/10.1175/1520-0493(2001)129<0550:iorthfv>2.0.co;2)
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. Retrieved from <https://arxiv.org/abs/1512.03385>
- Hersbach, H. (2000). Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, 15(5), 559–570. <https://doi.org/10.1175/1520-0434>
- Hess, P., & Boers, N. (2022). Deep learning for improving numerical weather prediction of heavy rainfall. *Journal of Advances in Modeling Earth Systems*, 14(3), e2021MS002765. <https://doi.org/10.1029/2021MS002765>
- Hewson, T. D., & Pillosu, F. M. (2021). A low-cost post-processing technique improves weather forecasts around the world. *Communications Earth & Environment*, 2(132), 1–10. <https://doi.org/10.1038/s43247-021-00185-9>

- Holden, Z. A., Abatzoglou, J. T., Luce, C. H., & Baggett, L. S. (2011). Empirical downscaling of daily minimum air temperature at very fine resolutions in complex terrain. *Agricultural and Forest Meteorology*, 151(8), 1066–1073. <https://doi.org/10.1016/j.agrformet.2011.03.011>
- Huang, X. (2020). Deep-learning based climate downscaling using the super-resolution method: A case study over the Western US. *Geoscientific Model Development Discussions*, 1–18. <https://doi.org/10.5194/gmd-2020-214>
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. Retrieved from <https://arxiv.org/abs/1412.6980>
- Klocek, S., Dong, H., Dixon, M., Kanengoni, P., Kazmi, N., Luferenko, P., et al. (2021). MS-nowcasting: Operational precipitation nowcasting with convolutional LSTMs at Microsoft Weather. In *NeurIPS 2021 Workshop on tackling climate change with machine learning*.
- Kumar, B., Chattopadhyay, R., Singh, M., Chaudhari, N., Kodari, K., & Barve, A. (2021). Deep learning–based downscaling of summer monsoon rainfall data over Indian region. *Theoretical and Applied Climatology*, 143(3), 1145–1156. <https://doi.org/10.1007/s00704-020-03489-6>
- Kurach, K., Lucic, M., Zhai, X., Michalski, M., & Gelly, S. (2018). A large-scale study on regularization and normalization in GANs. Retrieved from <https://arxiv.org/abs/1807.04720>
- Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 105–114). <https://doi.org/10.1109/CVPR.2017.19>
- Leinonen, J., Nerini, D., & Berne, A. (2020). Stochastic super-resolution for downscaling time-evolving atmospheric fields with a generative adversarial network. *IEEE Transactions on Geoscience and Remote Sensing*, 59(9), 7211–7223. <https://doi.org/10.1109/TGRS.2020.3032790>
- Lin, G., Wu, Q., Huang, X., Qiu, L., & Chen, X. (2017). Deep convolutional networks-based image super-resolution. In D.-S. Huang, V. Bevilacqua, P. Premaratne, & P. Gupta (Eds.), *Intelligent computing theories and application* (pp. 338–344). Springer International Publishing. https://doi.org/10.1007/978-3-319-63309-1_31
- Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *ICML workshop on deep learning for audio, speech and language processing*.
- Matheson, J. E., & Winkler, R. L. (1976). Scoring rules for continuous probability distributions. *Management Science*, 22(10), 1087–1096. <https://doi.org/10.1287/mnsc.22.10.1087>
- Met Office. (2003). 1 km resolution UK composite rainfall data from the Met Office Nimrod system. NCAS British atmospheric data centre. Retrieved from <https://catalogue.ceda.ac.uk/uuid/27dd6fba67f667a18c62de5c3456350>
- Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. Retrieved from <https://arxiv.org/abs/1411.1784>
- Palmer, T. (2020a). A vision for numerical weather prediction in 2030. Retrieved from <https://doi.org/10.48550/ARXIV.2007.04830> <https://arxiv.org/abs/2007.04830>
- Palmer, T. (2020b). White paper one contributor: Tim Palmer. Retrieved from <https://ppe-openplatform.wmo.int/en/WP1TP>
- Price, I., & Rasp, S. (2022). Increasing the accuracy and resolution of precipitation forecasts using deep generative models. Retrieved from <https://arxiv.org/abs/2203.12297>
- Pulkkinen, S., Nerini, D., Pérez Hortal, A. A., Velasco-Forero, C., Seed, A., Germann, U., & Foresti, L. (2019). Pysteps: An open-source Python library for probabilistic precipitation nowcasting (v1.0). *Geoscientific Model Development*, 12(10), 4185–4219. <https://doi.org/10.5194/gmd-12-4185-2019>
- Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., et al. (2021). Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878), 672–677. <https://doi.org/10.1038/s41586-021-03854-z>
- Rebora, N., Ferraris, L., von Hardenberg, J., & Provenzale, A. (2006). RainFARM: Rainfall downscaling by a filtered autoregressive model. *Journal of Hydrometeorology*, 7(4), 724–738. <https://doi.org/10.1175/JHM517.1>
- Roberts, N. (2008). Assessing the spatial and temporal variation in the skill of precipitation forecasts from an NWP model. *Meteorological Applications*, 15(1), 163–169. <https://doi.org/10.1002/met.57>
- Roberts, N. M., & Lean, H. W. (2008). Scale-selective verification of rainfall accumulations from high-resolution forecasts of convective events. *Monthly Weather Review*, 136(1), 78–97. <https://doi.org/10.1175/2007MWR2123.1>
- Rossa, A., Nurmi, P., & Ebert, E. (2008). Overview of methods for the verification of quantitative precipitation forecasts. In S. Michaelides (Ed.), *Precipitation: Advances in measurement, estimation and prediction* (pp. 419–452). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-77655-0_16
- Ruzanski, E., & Chandrasekar, V. (2011). Scale filtering for improved nowcasting performance in a high-resolution X-band radar network. *IEEE Transactions on Geoscience and Remote Sensing*, 49(6), 2296–2307. <https://doi.org/10.1109/TGRS.2010.2103946>
- Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One*, 10(3), 1–21. <https://doi.org/10.1371/journal.pone.0118432>
- Sha, Y., Gagne, D. J., II, West, G., & Stull, R. (2020). Deep-learning-based gridded downscaling of surface meteorological variables in complex terrain. Part II: Daily precipitation. *Journal of Applied Meteorology and Climatology*, 59(12), 2075–2092. <https://doi.org/10.1175/JAMC-D-20-0058.1>
- Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-k., & Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems* (Vol. 1, pp. 802–810). MIT Press.
- Sønderby, C. K., Espenholt, L., Heek, J., Dehghani, M., Oliver, A., Salimans, T., et al. (2020). MetNet: A neural weather model for precipitation forecasting. Retrieved from <https://arxiv.org/abs/2003.12140>
- Turkowsky, K. (1990). Filters for common resampling tasks. In A. S. Glassner (Ed.), *Graphics gems* (pp. 147–165). Morgan Kaufmann. <https://doi.org/10.1016/B978-0-08-050753-8.50042-5>
- Vaughan, A., Tebbutt, W., Hosking, J. S., & Turner, R. E. (2022). Convolutional conditional neural processes for local climate downscaling. *Geoscientific Model Development*, 15(1), 251–268. <https://doi.org/10.5194/gmd-15-251-2022>
- Wang, F., Tian, D., Lowe, L., Kalin, L., & Lehrter, J. (2021). Deep learning for daily precipitation and temperature downscaling. *Water Resources Research*, 57(4), e2020WR029308. <https://doi.org/10.1029/2020WR029308>
- Wang, Z., Simoncelli, E. P., & Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers*, 2, 1398–1402. <https://doi.org/10.1109/ACSSC.2003.1292216>
- Watson, C. D., Wang, C., Lynar, T., & Weldemariam, K. (2020). Investigating two super-resolution methods for downscaling precipitation: ESRGAN and CAR. Retrieved from <https://arxiv.org/abs/2012.01233>
- Yu, W., Nakakita, E., Kim, S., & Yamaguchi, K. (2016). Impact assessment of uncertainty propagation of ensemble NWP rainfall to flood forecasting with catchment scale. *Advances in Meteorology*, 2016, 1–17. <https://doi.org/10.1155/2016/1384302>