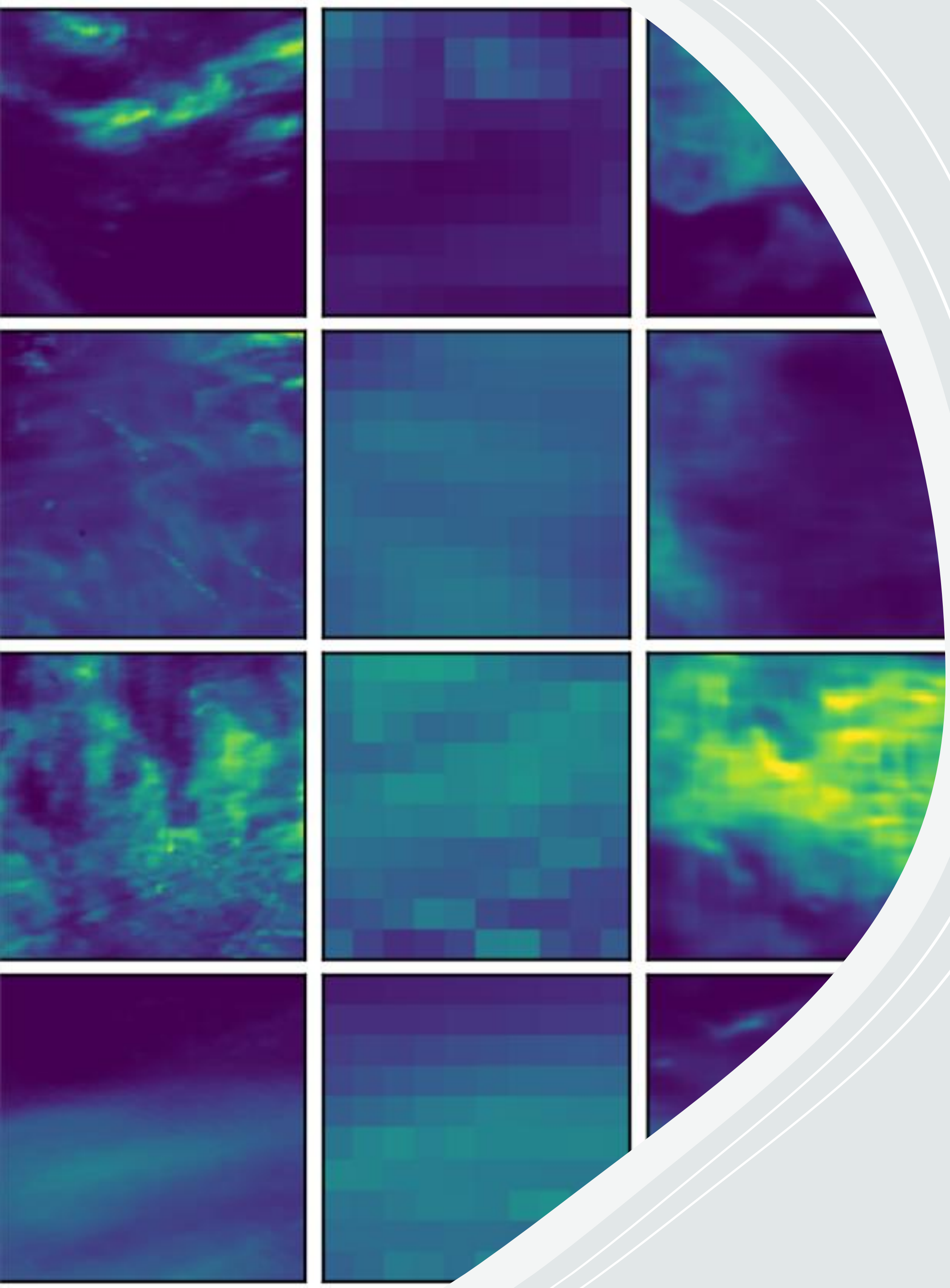


A Generative Deep Learning Approach to Stochastic Downscaling of Precipitation Forecasts

Lucy Harris & Andrew McRae, AOPP, University of Oxford
Mat Chantry, ECMWF

Peter Dueben, ECMWF
Tim Palmer, AOPP

Problem and motivation

Global forecast models produce good large-scale predictions, but output cannot be used directly for small spatial scales (<1km)

Precipitation is particularly variable over small lengthscales (c.f. pressure, wind, ⋯)

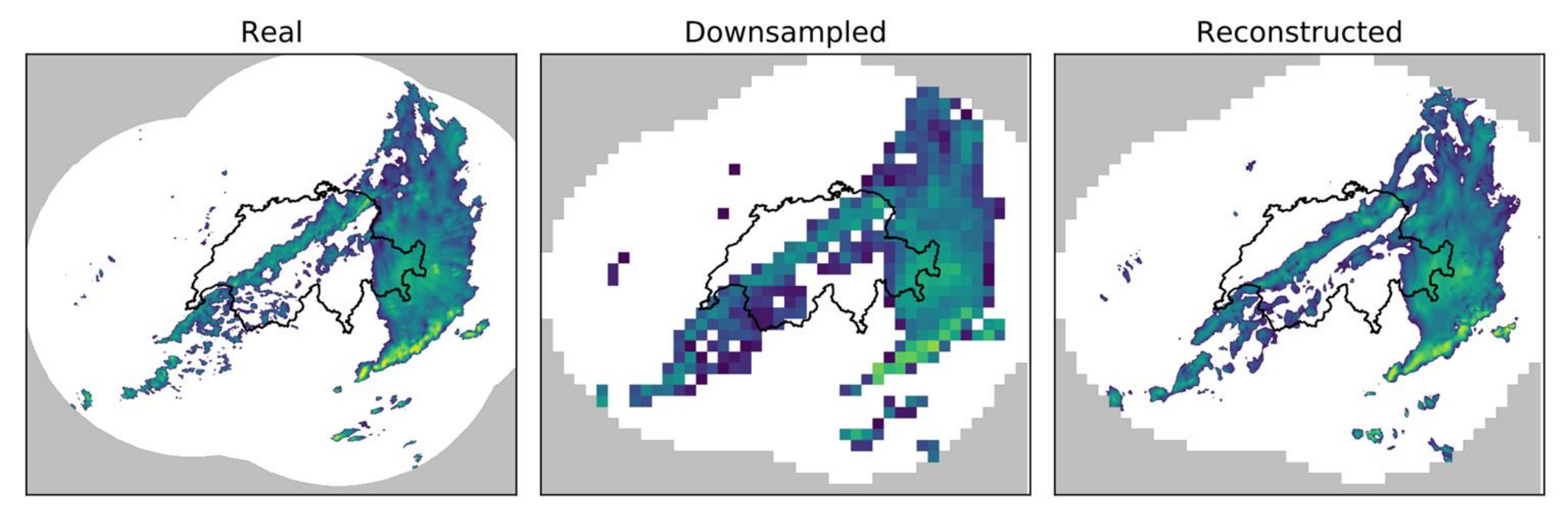
Current workflows include further postprocessing, running LAMs, etc.

Post-processing downscaling of precipitation forecasts is necessary to assess the impact of extreme rainfall situations

Can ML models help? What are the limits of postprocessing global model output?

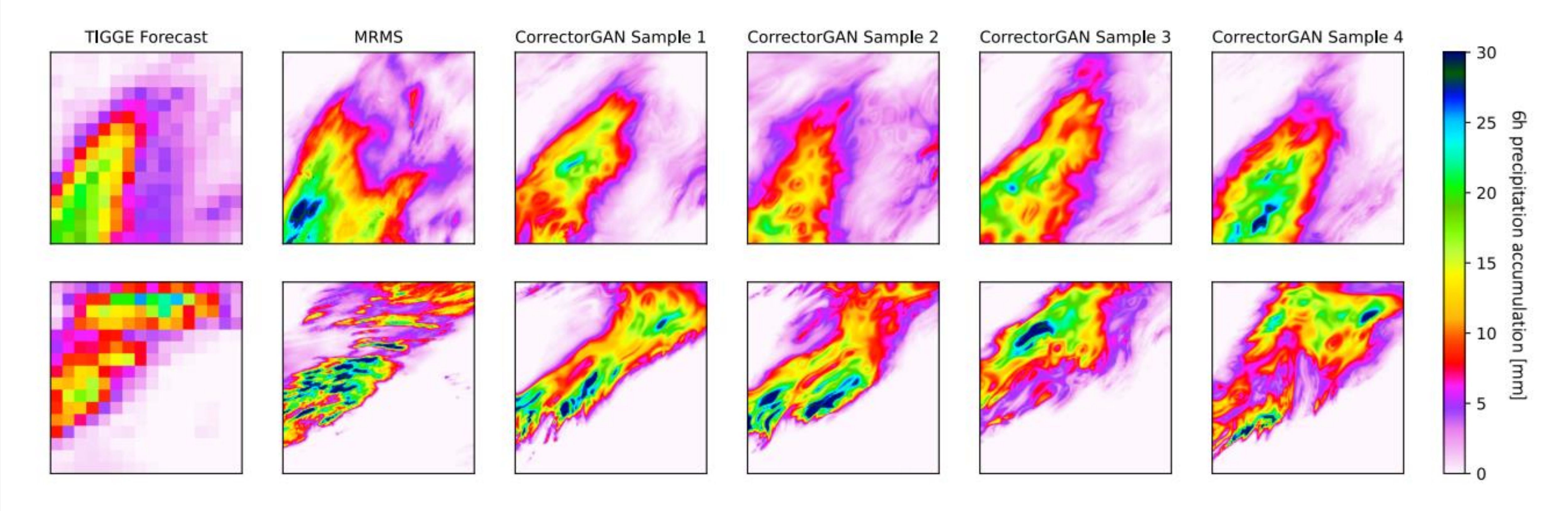
Stochastic Super-Resolution for Downscaling **<u>Time-Evolving</u>** Atmospheric Fields with a GAN

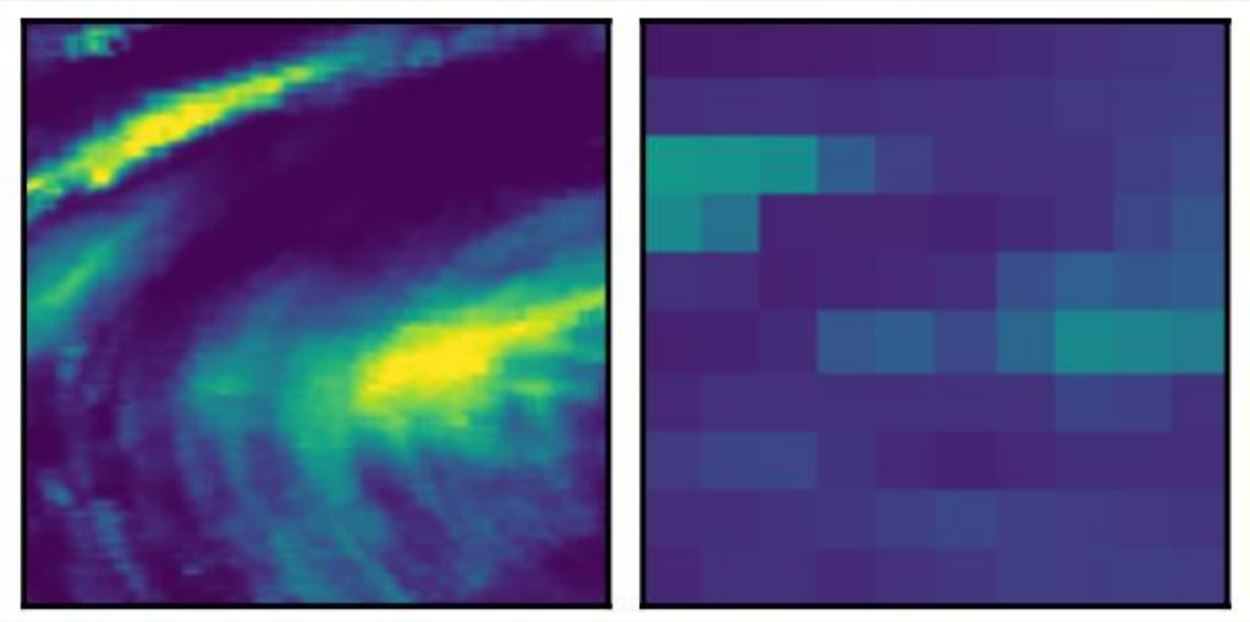
Leinonen et al. 2020



Increasing the accuracy and resolution of precipitation forecasts using deep generative models

Price & Rasp 2022





NIMROD
target

IFS
input

1km C-band radar-based rainfall, adjusted with gauge measurements. Regridded to 0.01°, accumulated to 1-hour.

Domain: 49.5 - 59 latitude, -7.5 - 2 longitude.

~9km forecast output (lead-times 6-18 hours)
Regridded to 0.1°, 1-hour.
Fields: Total precipitation, convective precipitation, CAPE, u-700hPa, v-700hPa, TOA incident solar radiation, Total column cloud liquid water, Total

inputs

Lo-res IFS fields:

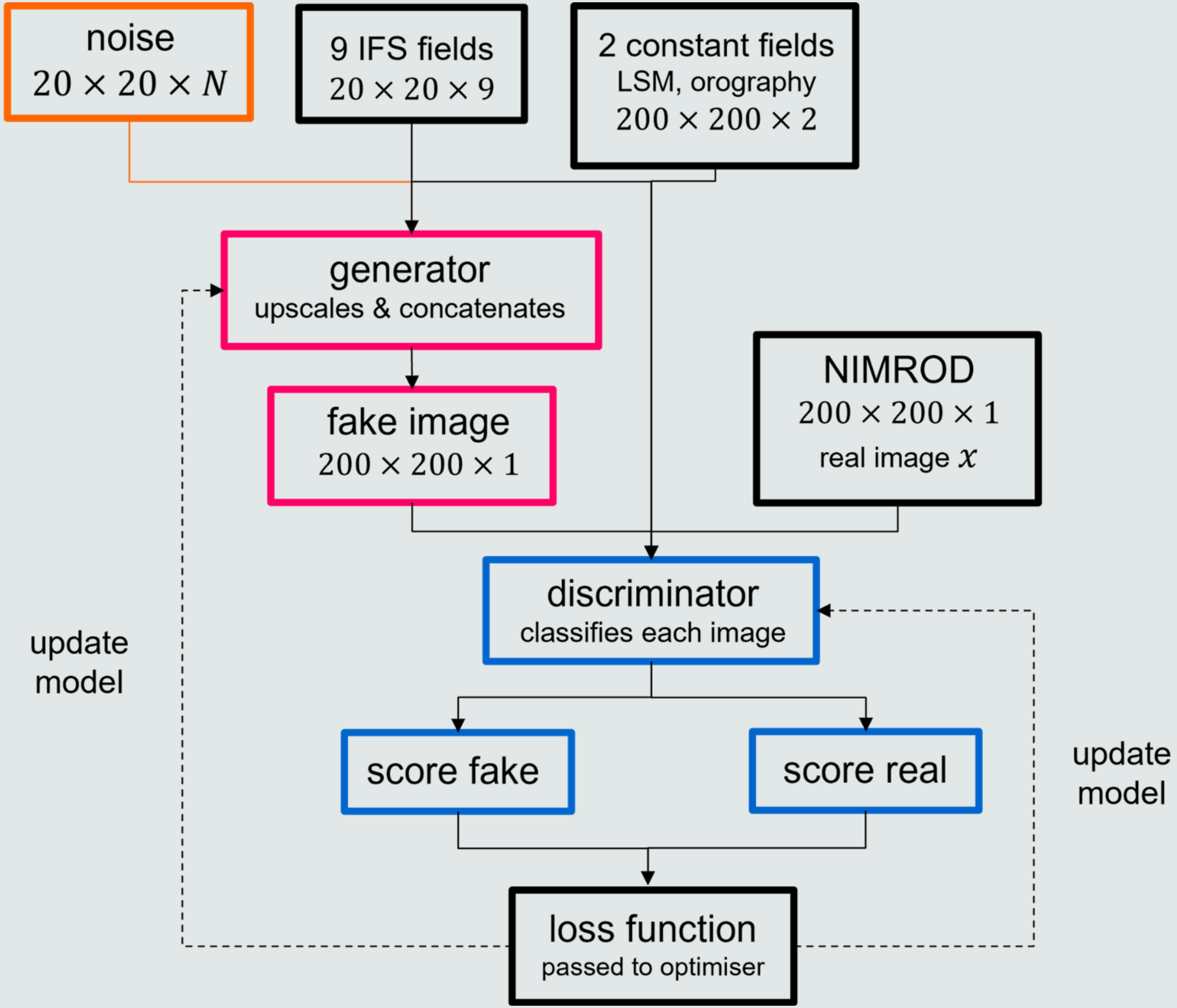
(based on ecPoint approach)

- Total precipitation
- Convective precipitation
- Surface pressure
- TOA incident solar radiation
- Convective available potential energy
- Total column cloud liquid water
- Total column water vapour
- u700 and v700

Hi-res ‘constant’ inputs:

- Orography
- Land-sea mask

cGAN


noise
$20 \times 20 \times N$
9 IFS fields
$20 \times 20 \times 9$
2 constant fields
LSM, orography
$200 \times 200 \times 2$
generator
upscales & concatenates
fake image
$200 \times 200 \times 1$
NIMROD
$200 \times 200 \times 1$
real image x
discriminator
classifies each image
score fake
score real
loss function
passed to optimiser
update model
update model

convolutional NN with residual blocks

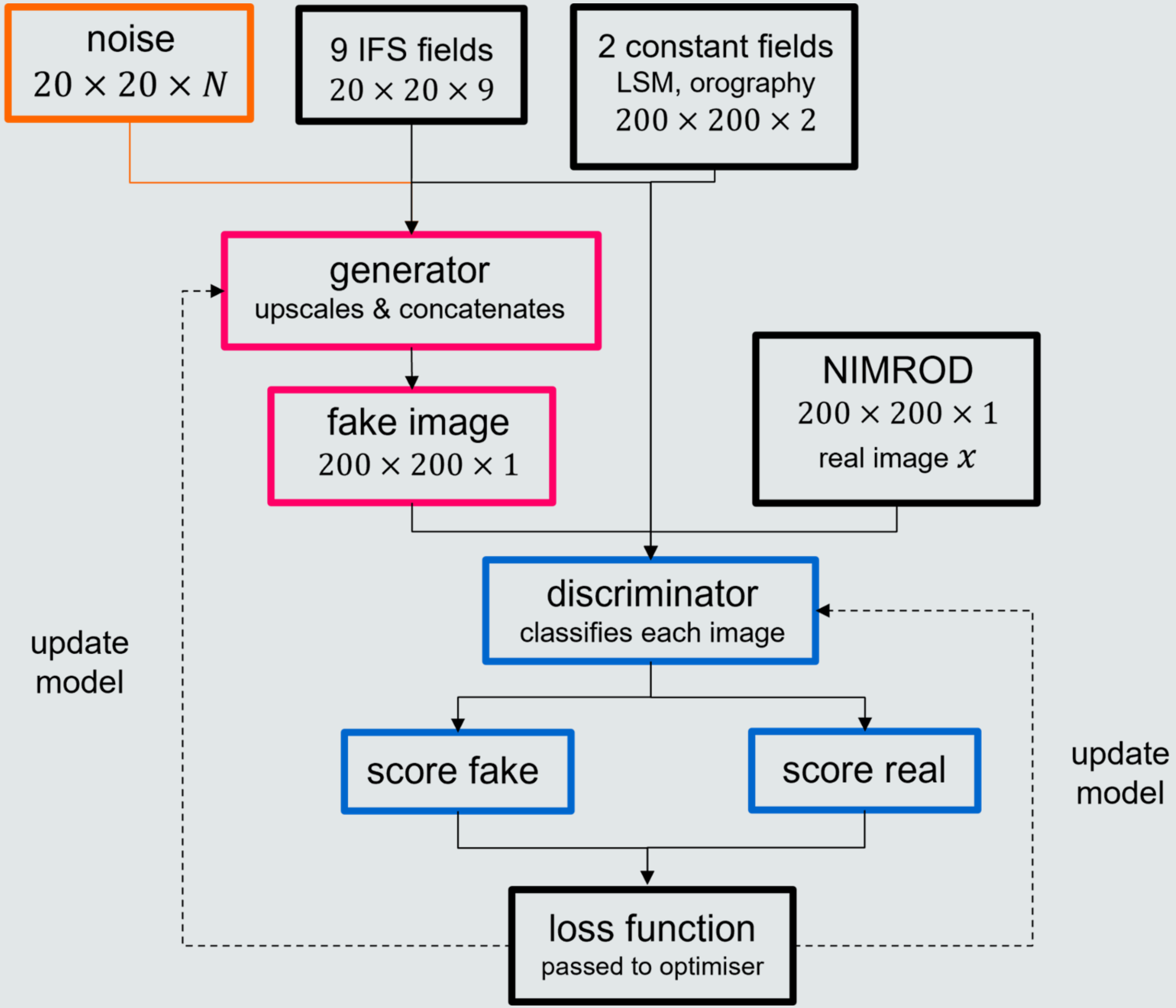
Wasserstein GAN with gradient penalty

trainable parameters

gen – 0.83 million (64 filters)

3.2 million (128 filters)

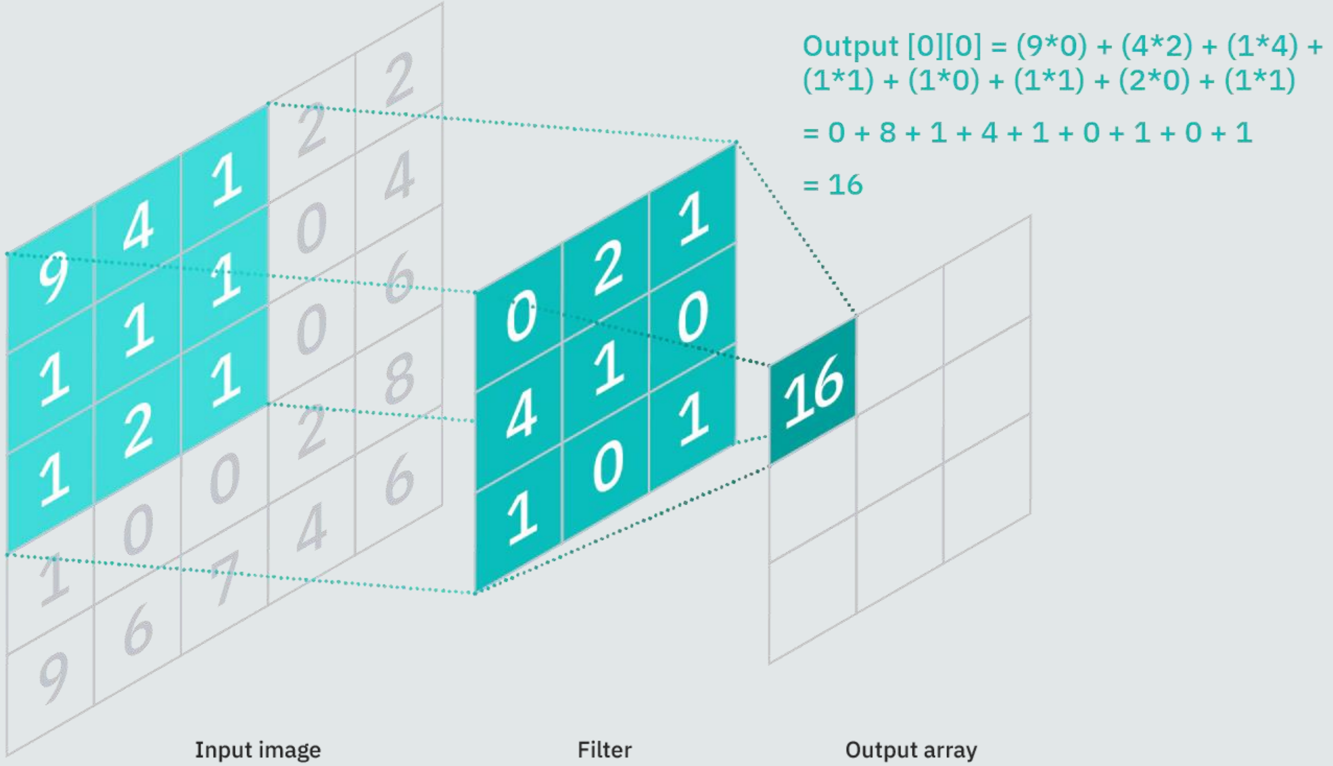
disc – 64.1 million (512 filters)



NN architecture

Convolutional layer:

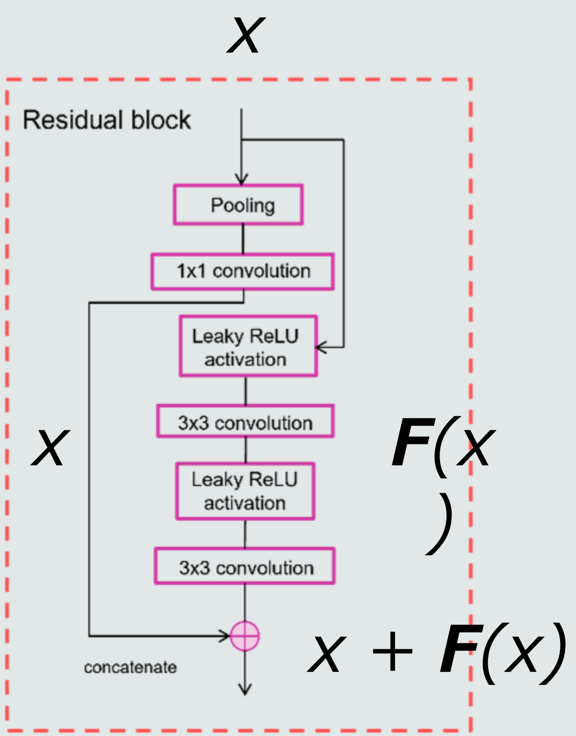
- a convolution is a linear operation: the multiplication of a set of weights with the input
- multiplication (dot product) is performed between an array of input data and a 2D array of weights called a filter
- the filter is smaller than the input and is applied systematically to each overlapping patch of the input data
- this allows the filter to detect features across the entire image
- Many filters used at once, e.g. 64, 128, 256



NN architecture

Residual block:

- skip-connection blocks
- learns residual functions with reference to the layer inputs
- $x \rightarrow x + \boldsymbol{F}(x)$
- $\boldsymbol{F}(x)$ based on two 3x3 convolutions

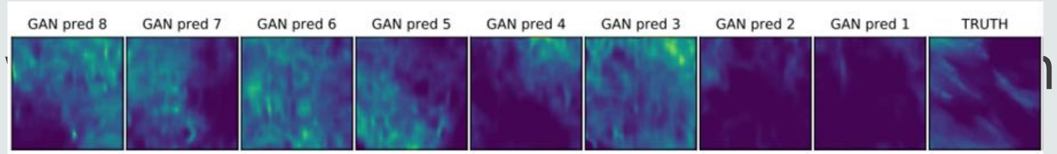


Content Loss term

- Used in Deepmind nowcasting paper
- *They borrowed it from earlier 'DVD-GAN' work, where it was crucial*

Idea: train generator on

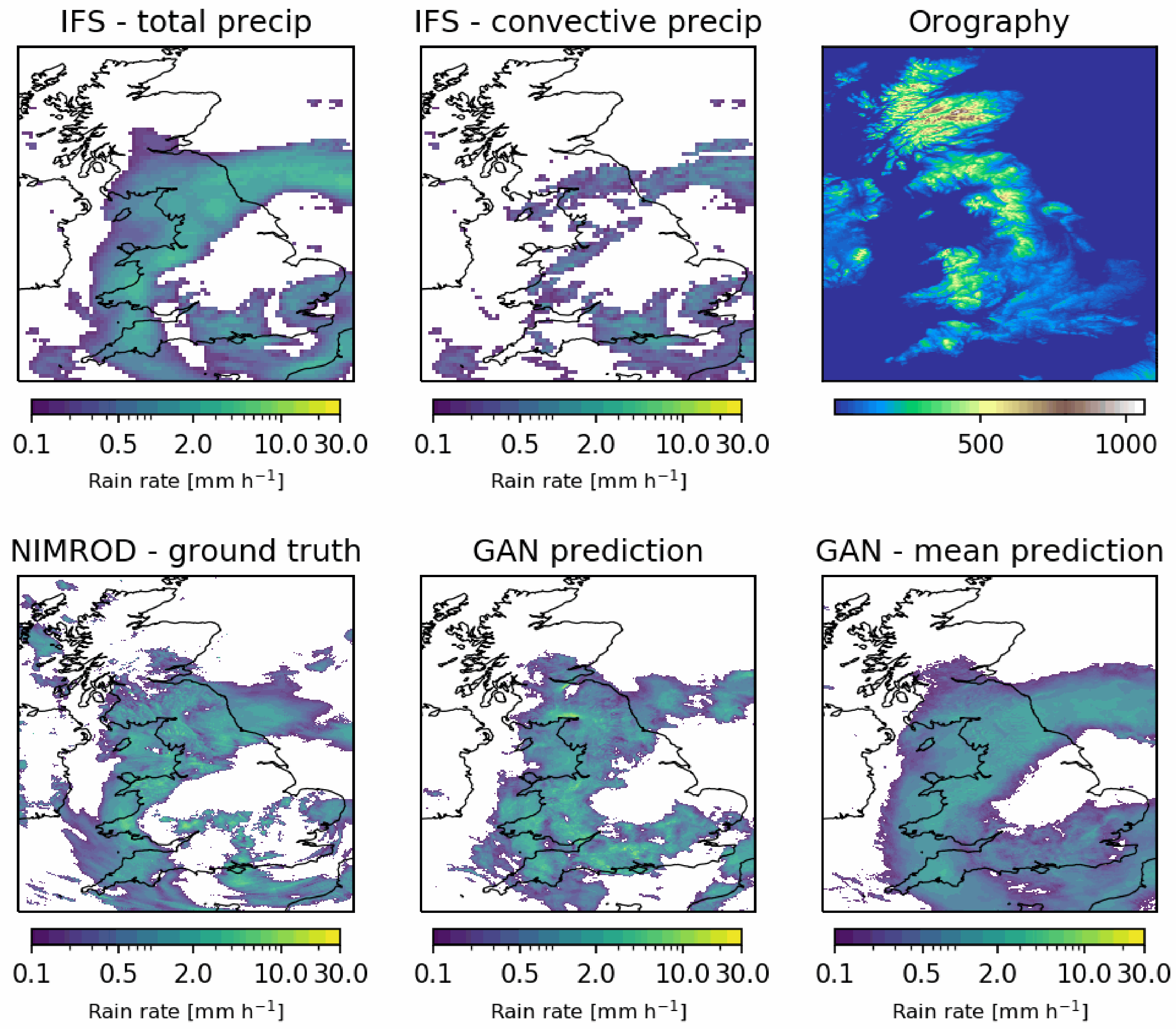
(discriminator loss) + …)



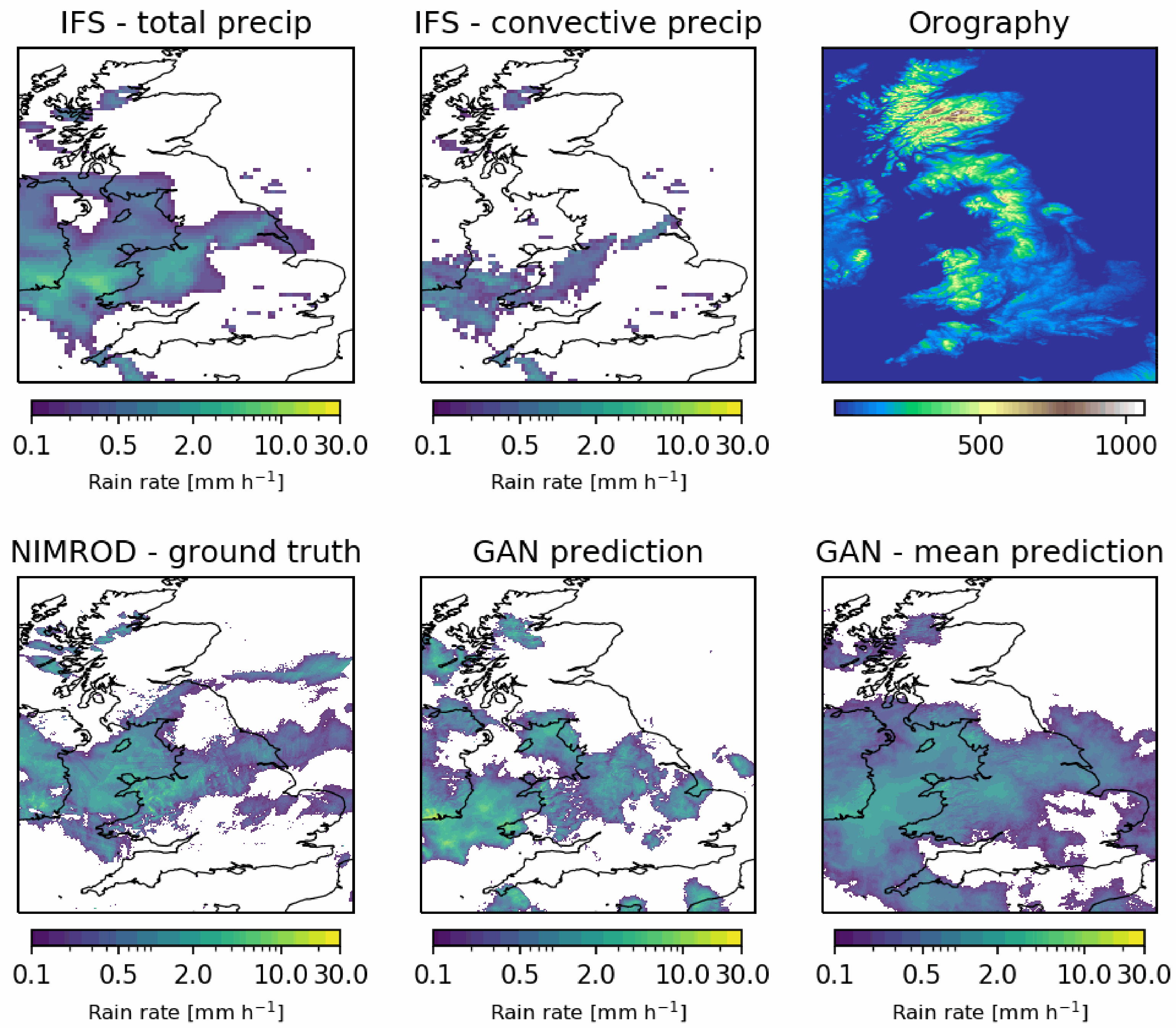
mean

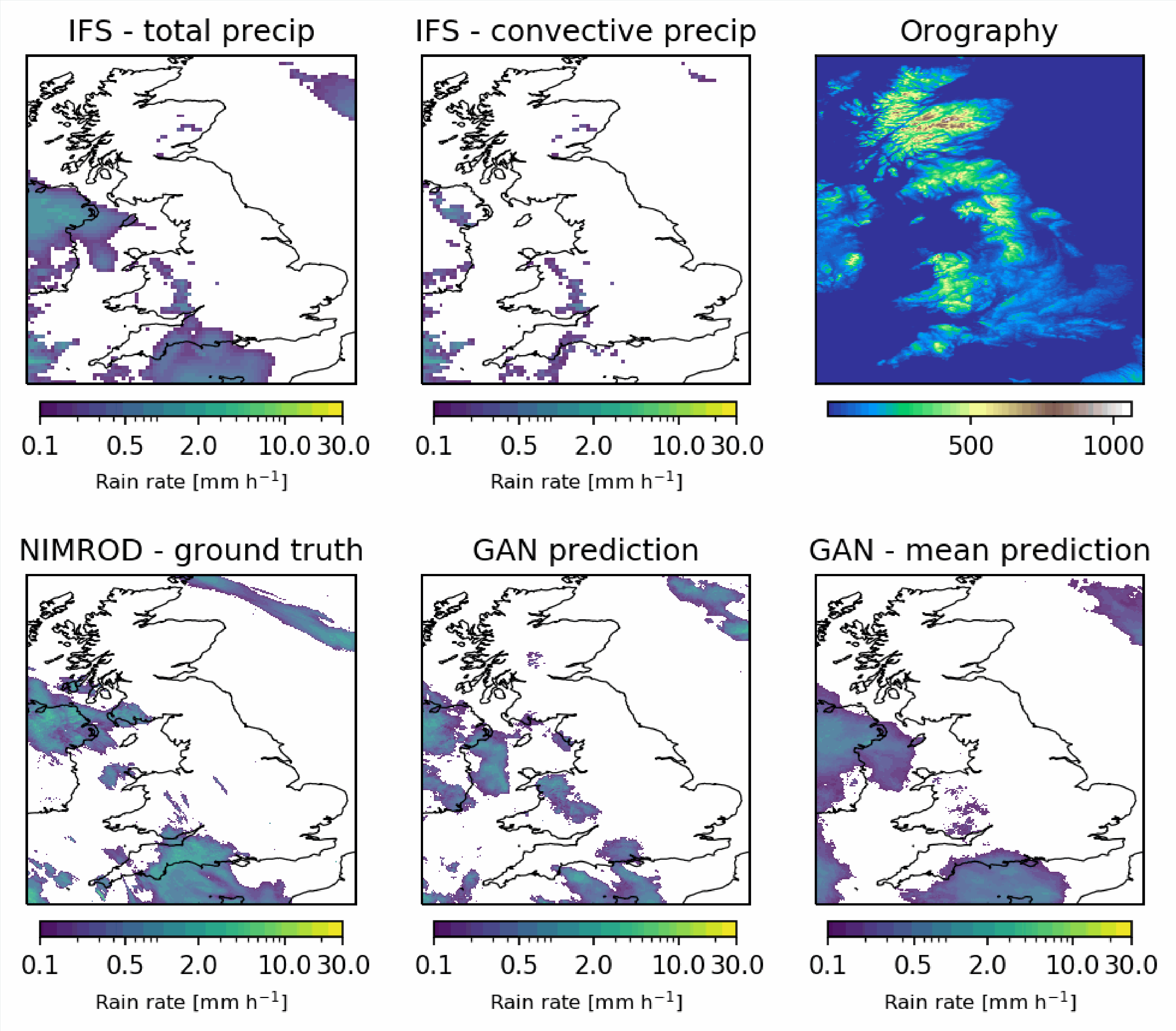
MSE

Results



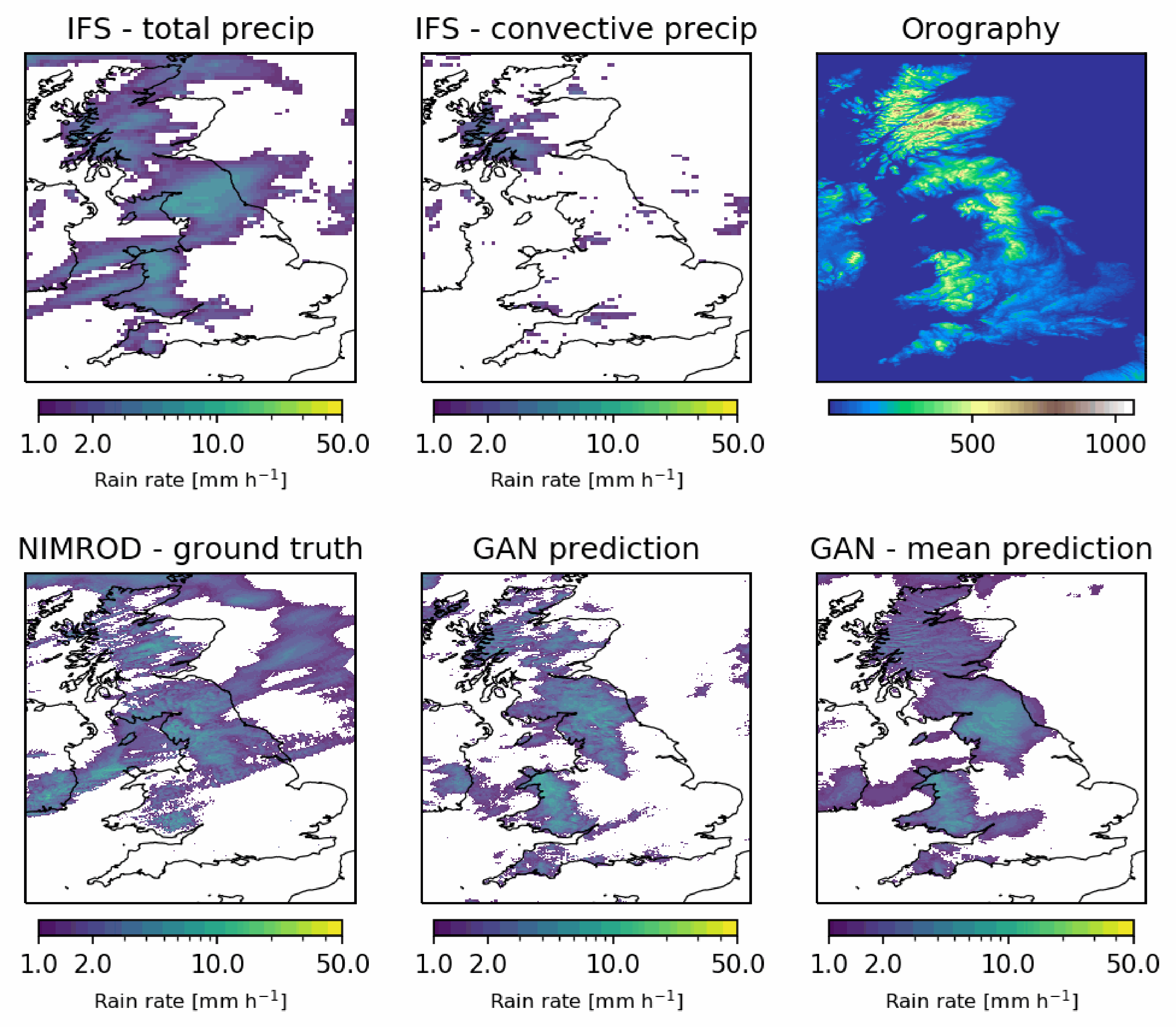
IFS - total precip
IFS - convective precip
Orography
0.1
0.5
2.0
10.0
30.0
Rain rate [mm h^{-1}]
500
1000
NIMROD - ground truth
GAN prediction
GAN - mean prediction


IFS - total precip
IFS - convective precip
Orography
0.1 0.5 2.0 10.0 30.0
Rain rate [mm h^{-1}]
500 1000
NIMROD - ground truth
GAN prediction
GAN - mean prediction


IFS - total precip
IFS - convective precip
Orography
0.1 0.5 2.0 10.0 30.0
Rain rate [mm h^{-1}]
0.1 0.5 2.0 10.0 30.0
Rain rate [mm h^{-1}]
500 1000
NIMROD - ground truth
GAN prediction
GAN - mean prediction
0.1 0.5 2.0 10.0 30.0
Rain rate [mm h^{-1}]
0.1 0.5 2.0 10.0 30.0
Rain rate [mm h^{-1}]
0.1 0.5 2.0 10.0 30.0
Rain rate [mm h^{-1}]

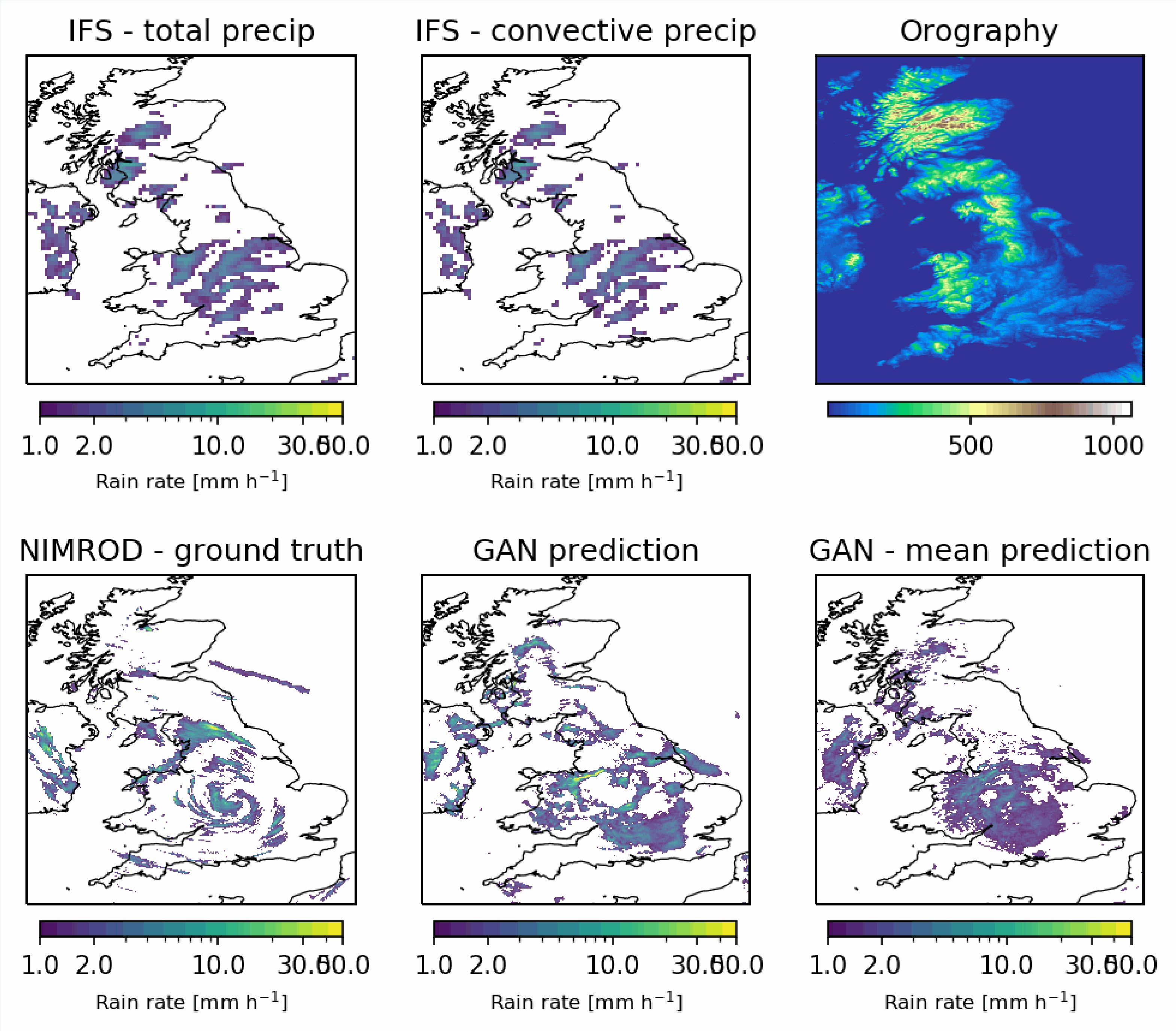
Extreme situation (Storm Ciara)

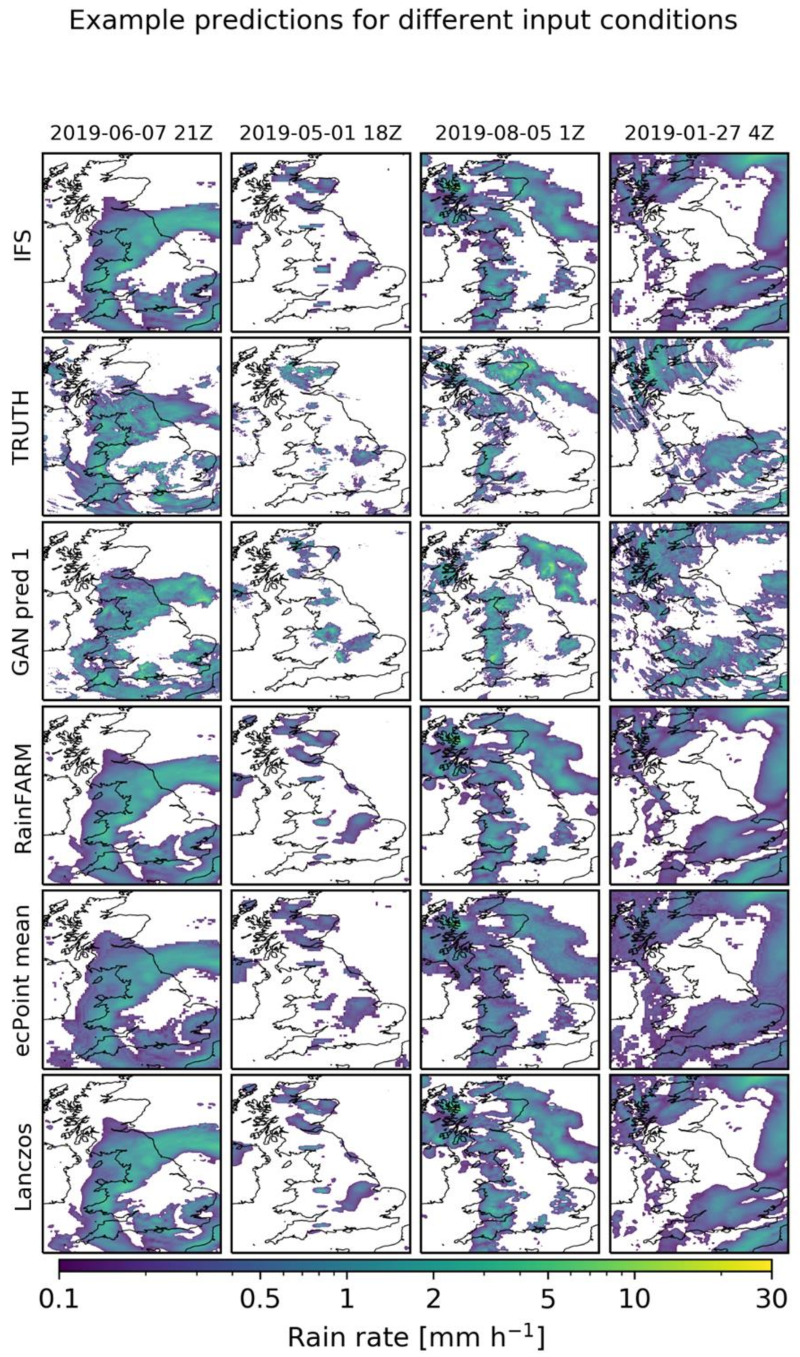
NB. change of colorbar from 0.1-30mm



Extreme situation (16:00-17:00 UTC 31/07/19)

NB. change of colorbar from 0.1-30mm





Example predictions for different input conditions
2019-06-07 21Z
2019-05-01 18Z
2019-08-05 1Z
2019-01-27 4Z
IFS
TRUTH
GAN pred 1
RainFARM
ecPoint mean
Lanczos
0.1
0.5
1
2
5
10
30
Rain rate [mm h^{-1}]

Quantitative metrics (256 full hourly images)

Model	Evaluation Metric						
	CRPS (mm/hr)					RALSD (dB)	RMSE (mm/hr)
	pixelwise	avg 4	max 4	avg 16	max 16		
GAN	**0.0883**	**0.0871**	**0.1180**	**0.0835**	**0.2166**	4.404	0.5953
VAE-GAN	0.0914	0.0904	0.1229	0.0875	0.2282	**4.276**	0.4890
ecPoint no-corr	0.0905	0.1077	0.4008	0.1195	1.5965	16.351	0.6445
ecPoint part-corr	0.0905	0.0897	0.1265	0.0887	0.2504	9.773	0.6445
RainFARM	0.1331	0.1332	0.1697	0.1286	0.2888	9.951	0.4441
Lanczos	0.1412	0.1392	0.1731	0.1309	0.2923	15.379	0.4470
Deterministic CNN	0.1347	0.1325	0.1644	0.1250	0.2817	16.738	**0.4042**

ecPoint = ecPoint approach, calibrated on training dataset for this problem

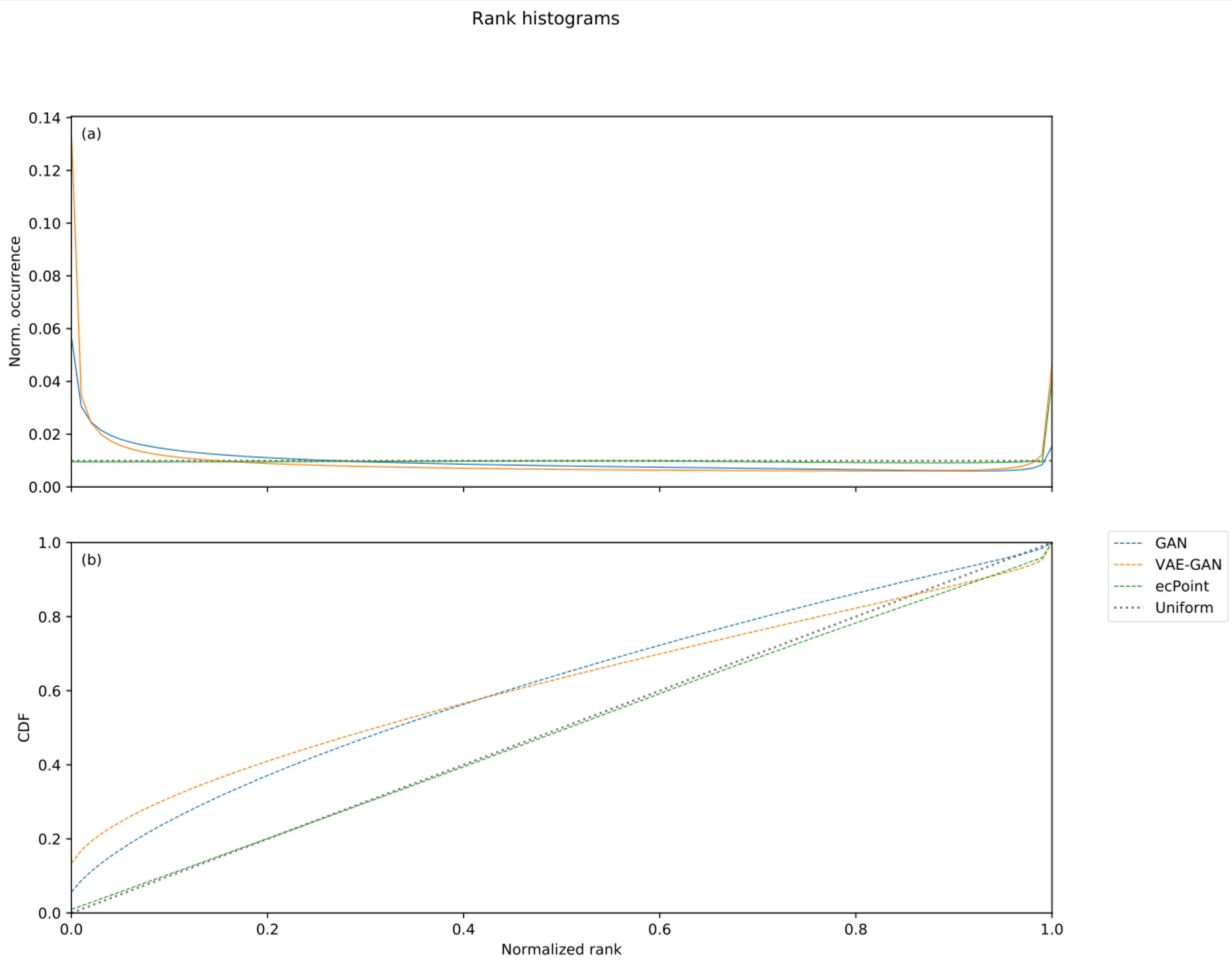
- no-corr, part-corr = naive methods of generating full images from ecPoint pixel data

RainFARM = stochastic method by Rebora et al. (2006), “extends power spectrum” but doesn’t handle forecast error

Lanczos = Lanczos interpolation

Deterministic CNN = neural network trained on MSE (not using GAN methodology)

Rank histograms plot (cGAN)



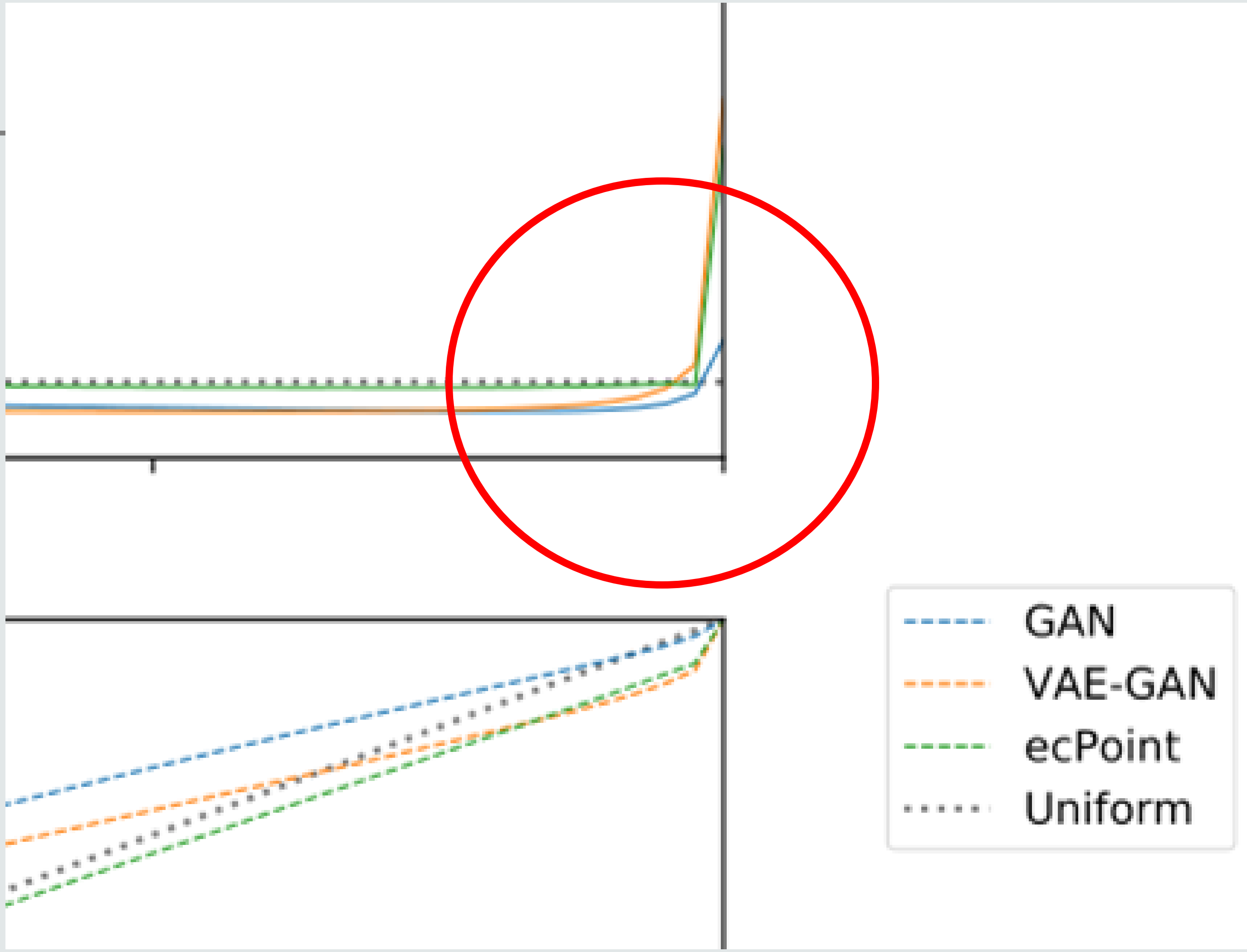
no. samples where pixel value is smaller than truth

$$r = \frac{N_s}{N_p}$$

rank

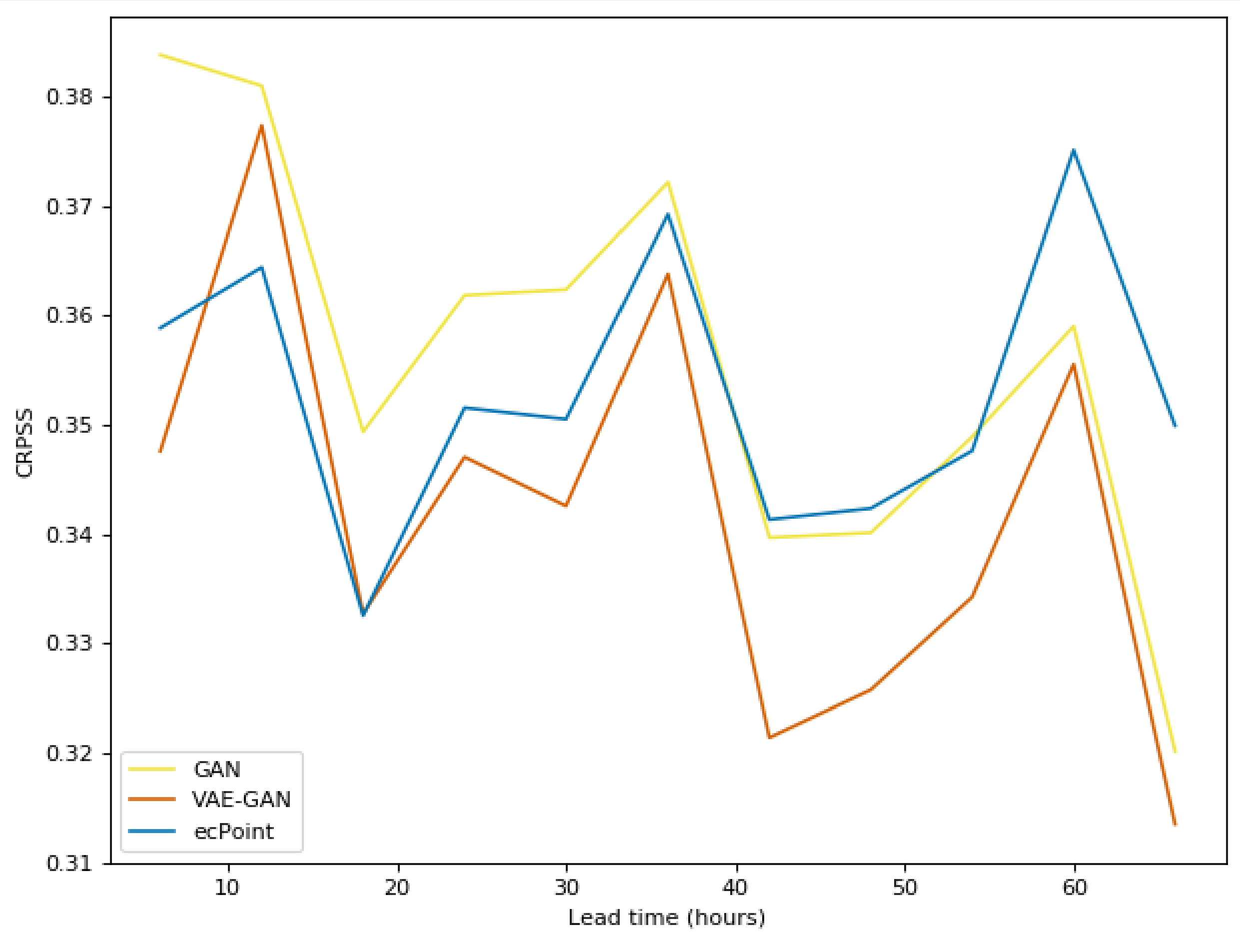
total no. predictions

Rank histograms plot (cGAN)



Lead time assessment

$\text{CRPSS} = 1 - \text{CRPS}_{fc} / \text{CRPS}_{bench}$



More results available···

Pooled CRPS scores

RALSD plots

ROC and precision–recall curves

Fractions Skill Scores

CRPS vs 0-72hr forecast lead time (without retraining on longer lead times)

Conclusions + Future Work

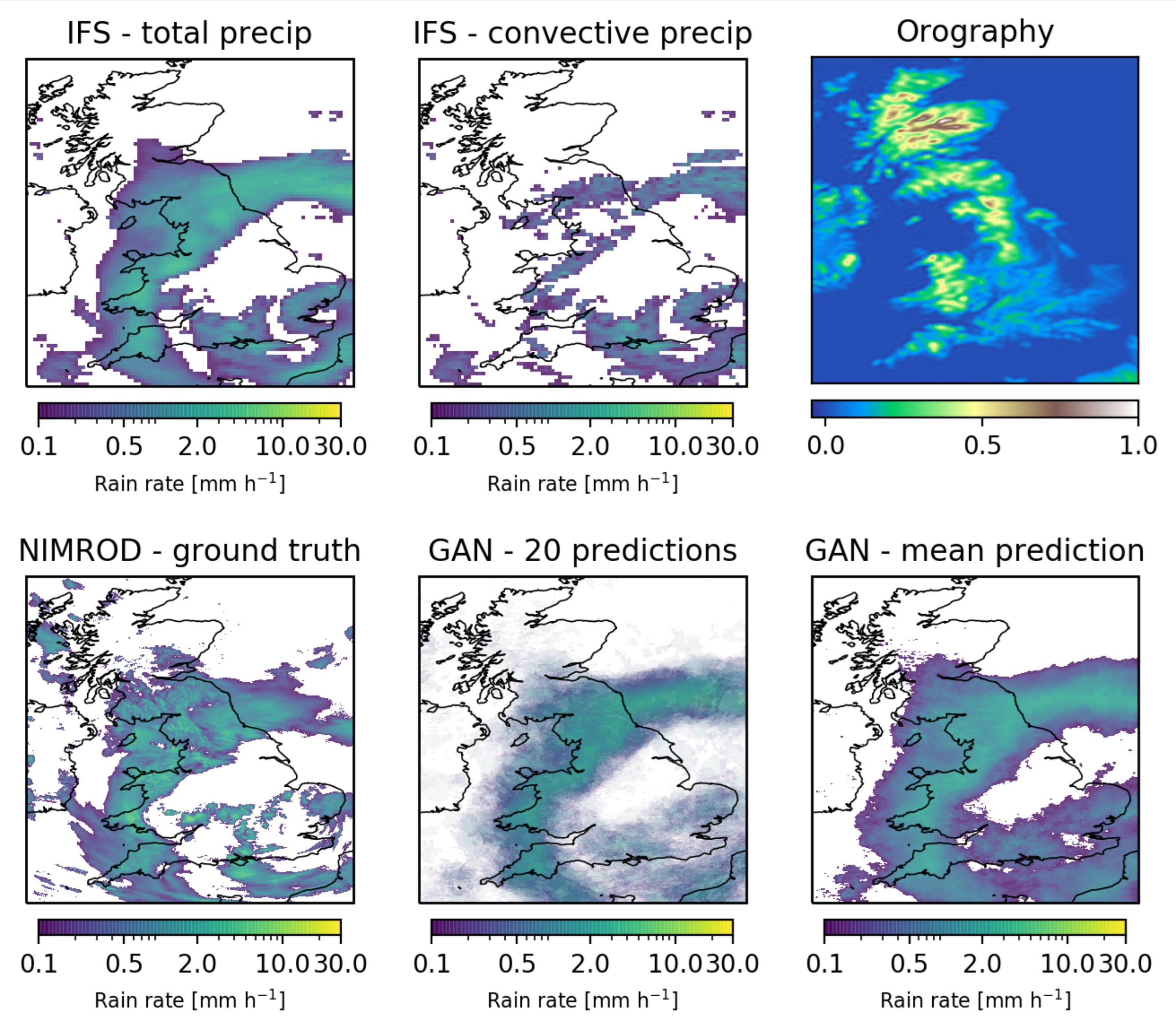
- GAN produces sharply varying but spatially coherent results
- Similar pixel-wise accuracy to ecPoint approach (better CRPS, worse calibration)
- Once trained, moderate computational cost (1s/sample on full image)

Future ideas:

- Temporally-consistent results
- Use ensemble information
- Incorporate into downstream hydrology model

questions

backup slides



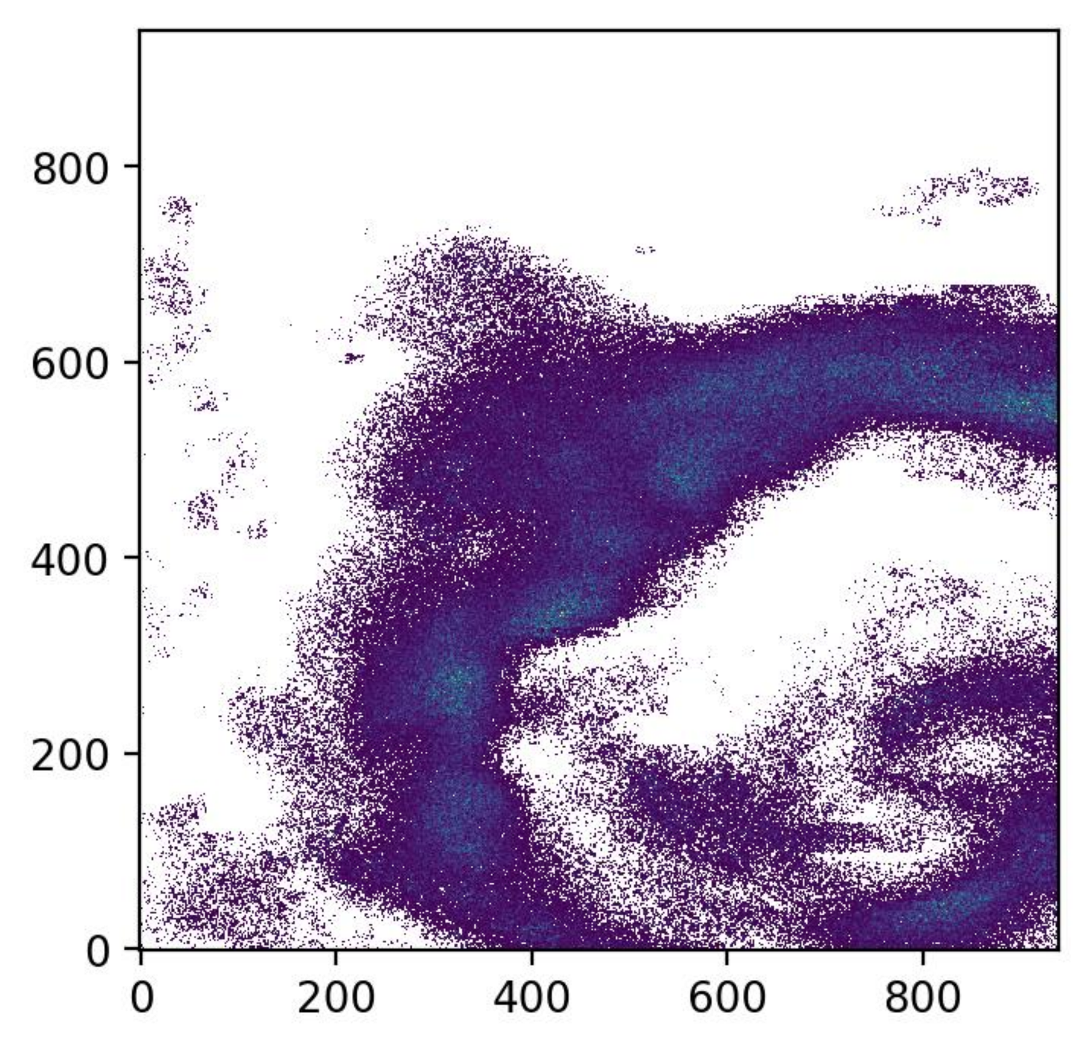
This uses old orography

Spatial coherer

Motivation: similar point-wise properties (CRPS, etc.) to ecPoint approach. Why is a spatially-coherent image better?

Two things we have tried:

- Pooling (used with CRPS, also ROC)
- Fractions Skill Score



ecPoint approach: each pixel sampled from parent IFS gridbox's PDF "ecPoint, no correlation"

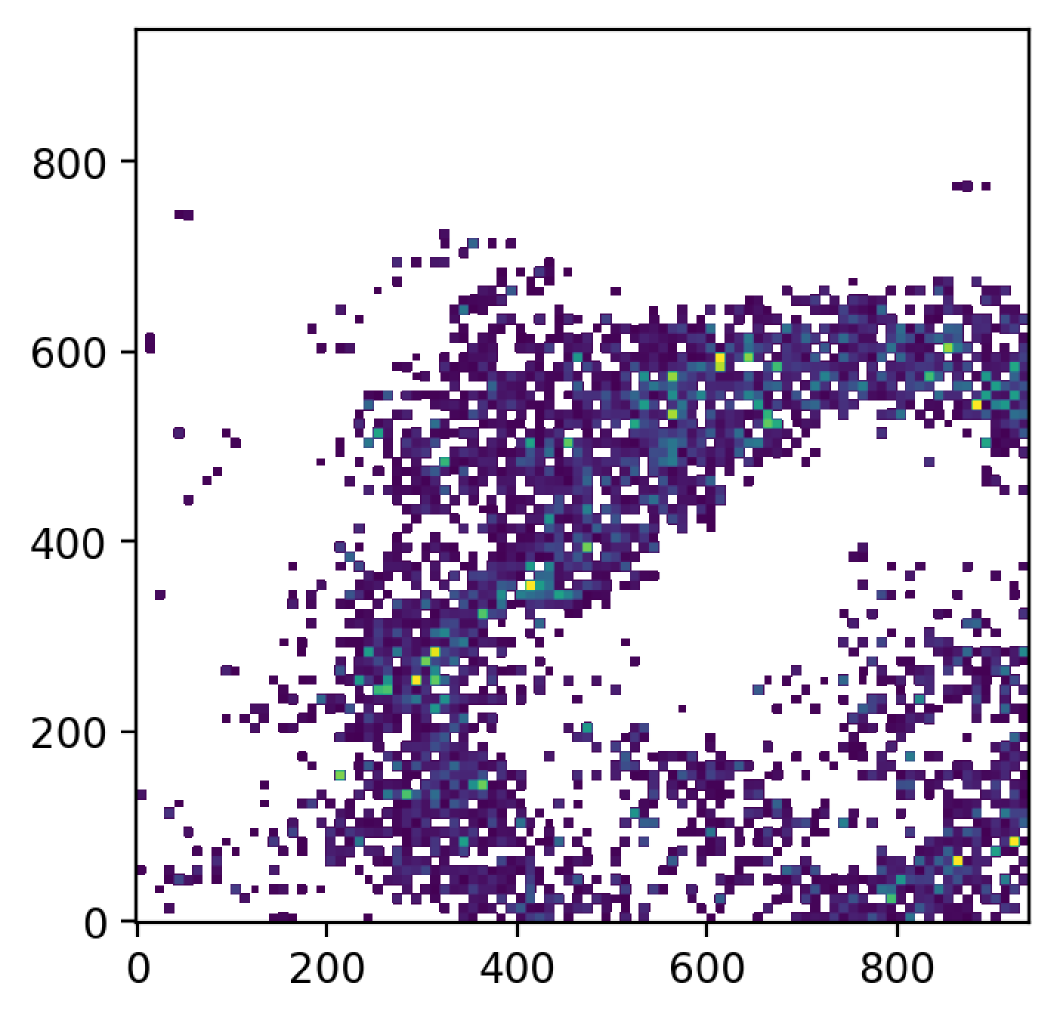
Spatial coherer

Motivation: similar point-wise properties to ecPoint approach. Why is a spatially-coherent image better?

Two things we have tried:

- Pooling (used with CRPS, also ROC)
- Fractions Skill Score

Also power spectrum…



ecPoint approach: each 10x10 block sampled from parent IFS gridbox's PDF "ecPoint part correlation"

Pooling

Used in Deepmind nowcasting paper

Idea: aggregate predictions and truth image over larger windows before calculating metric

- Average-pooling (~cumulative rainfall over a region)
- Max-pooling (max rainfall nearby)
- 4x4 and 16x16 windows

‘Poor man’s Fractions Skill Score’?

Fractions Skill Score

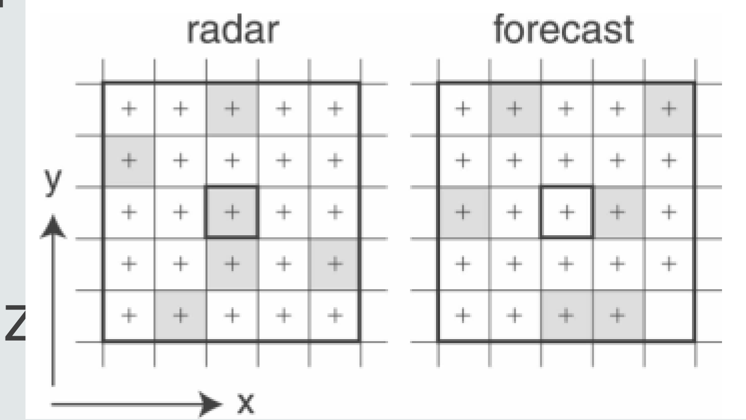
Original motivation: “on what spatial scale is a prediction skillful?”

1. For a precipitation threshold, binarise image
2. Average pred … v siz
3. Compute MSE
4. C

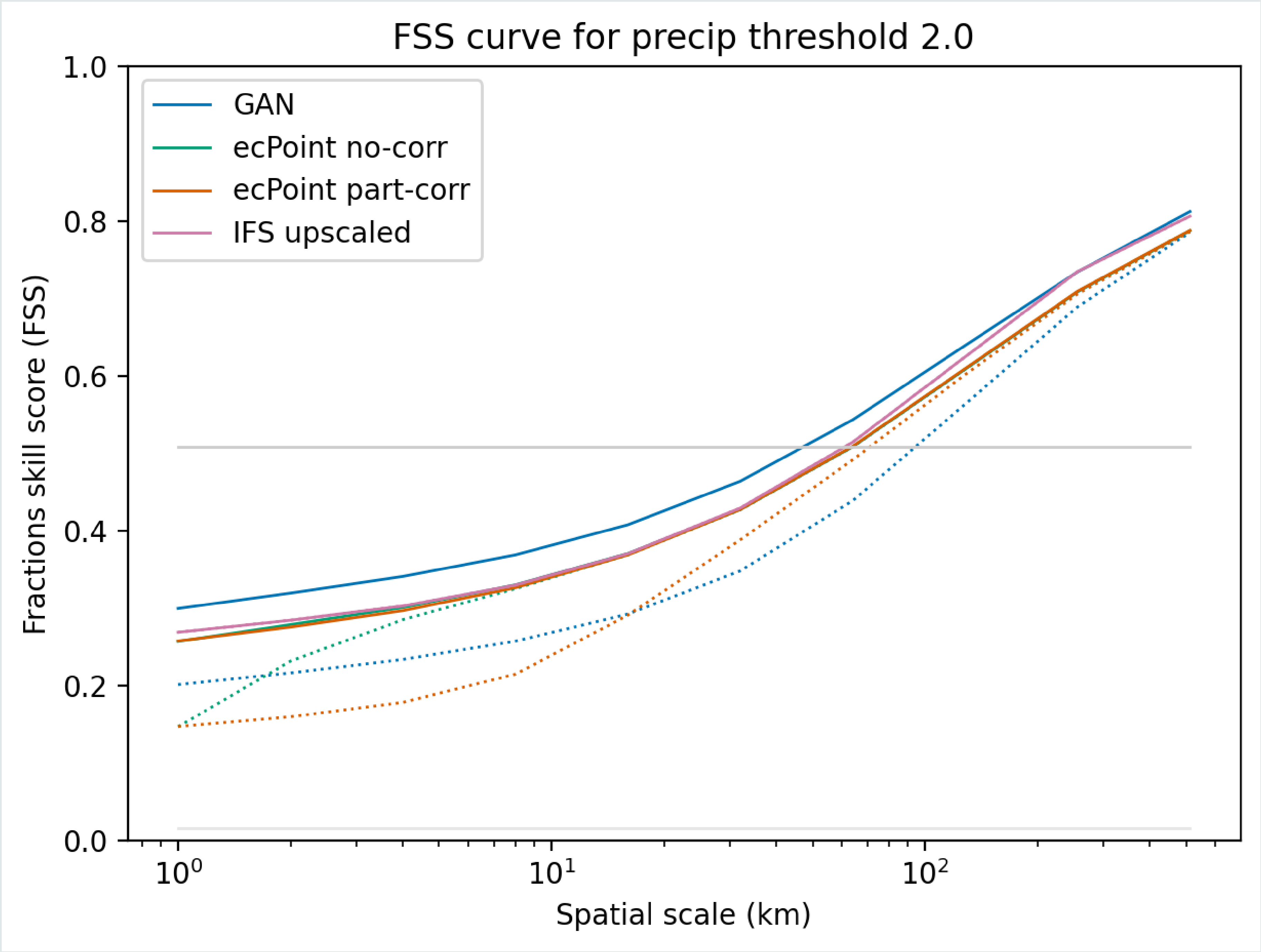
$$\mathrm{MSE}_{(n)} = \frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left[O_{(n)i,j} - M_{(n)i,j}\right]^2.$$

$$\mathrm{FSS}_{(n)} = \frac{\mathrm{MSE}_{(n)} - \mathrm{MSE}_{(n)\mathrm{ref}}}{\mathrm{MSE}_{(n)\mathrm{perfect}} - \mathrm{MSE}_{(n)\mathrm{ref}}} = 1 - \frac{\mathrm{MSE}_{(n)}}{\mathrm{MSE}_{(n)\mathrm{ref}}},$$

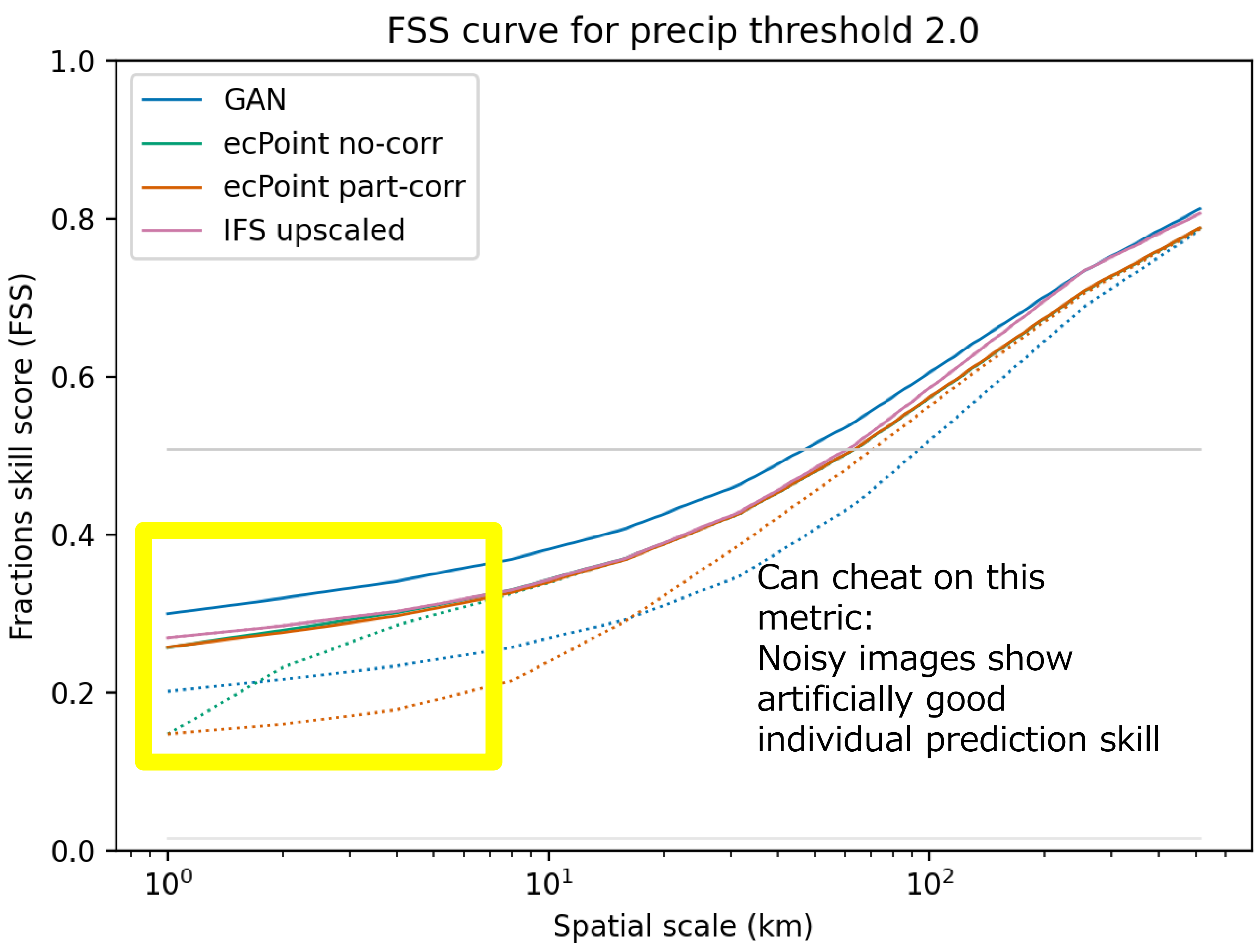
$$\mathrm{MSE}_{(n)\mathrm{ref}} = \frac{1}{N_x N_y}\left[\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} O_{(n)i,j}^2 + \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} M_{(n)i,j}^2\right].$$



Have generalised into ensemble version too.

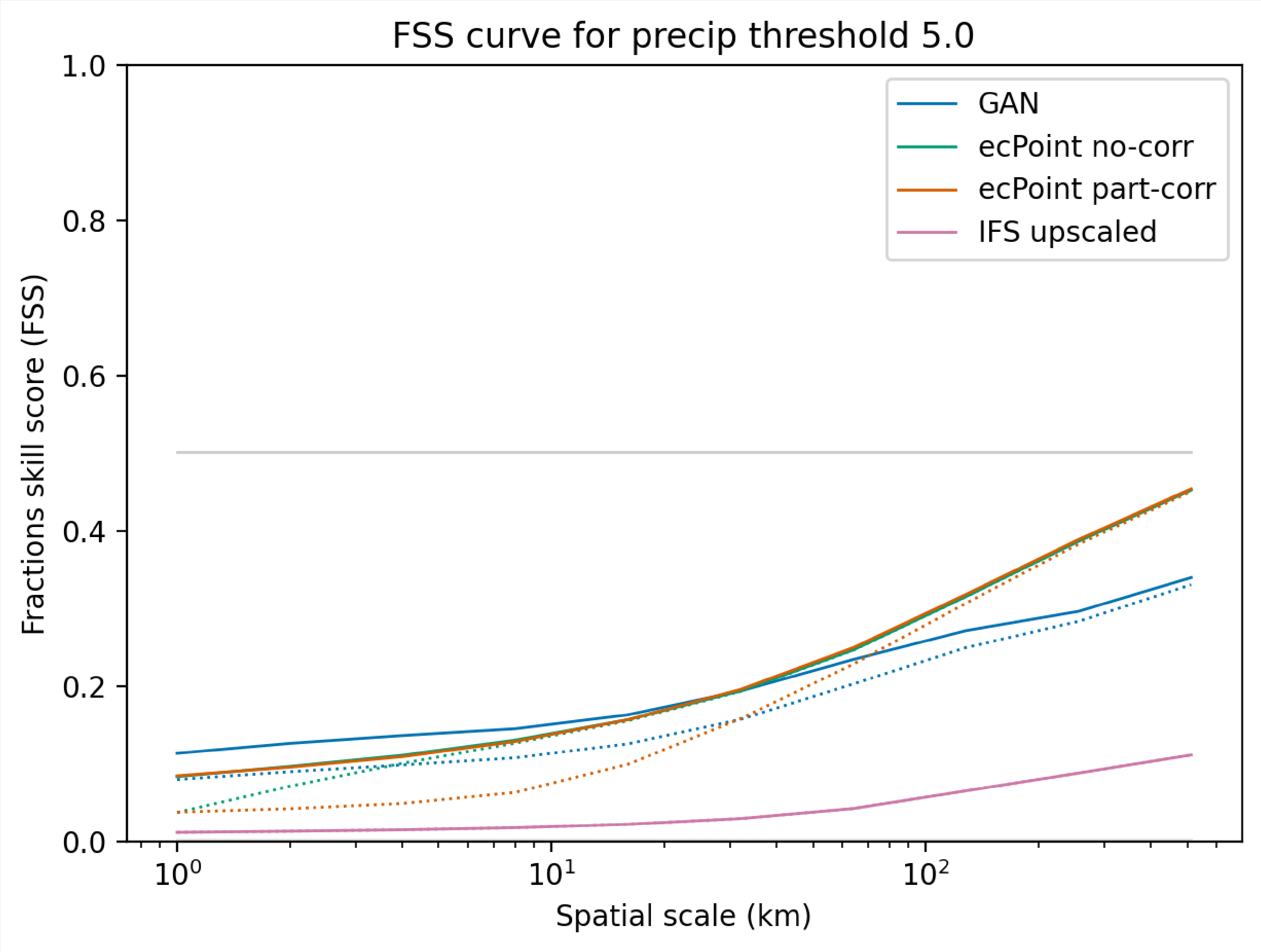


Solid = “ensemble skill”
Dotted = “individual prediction skill”

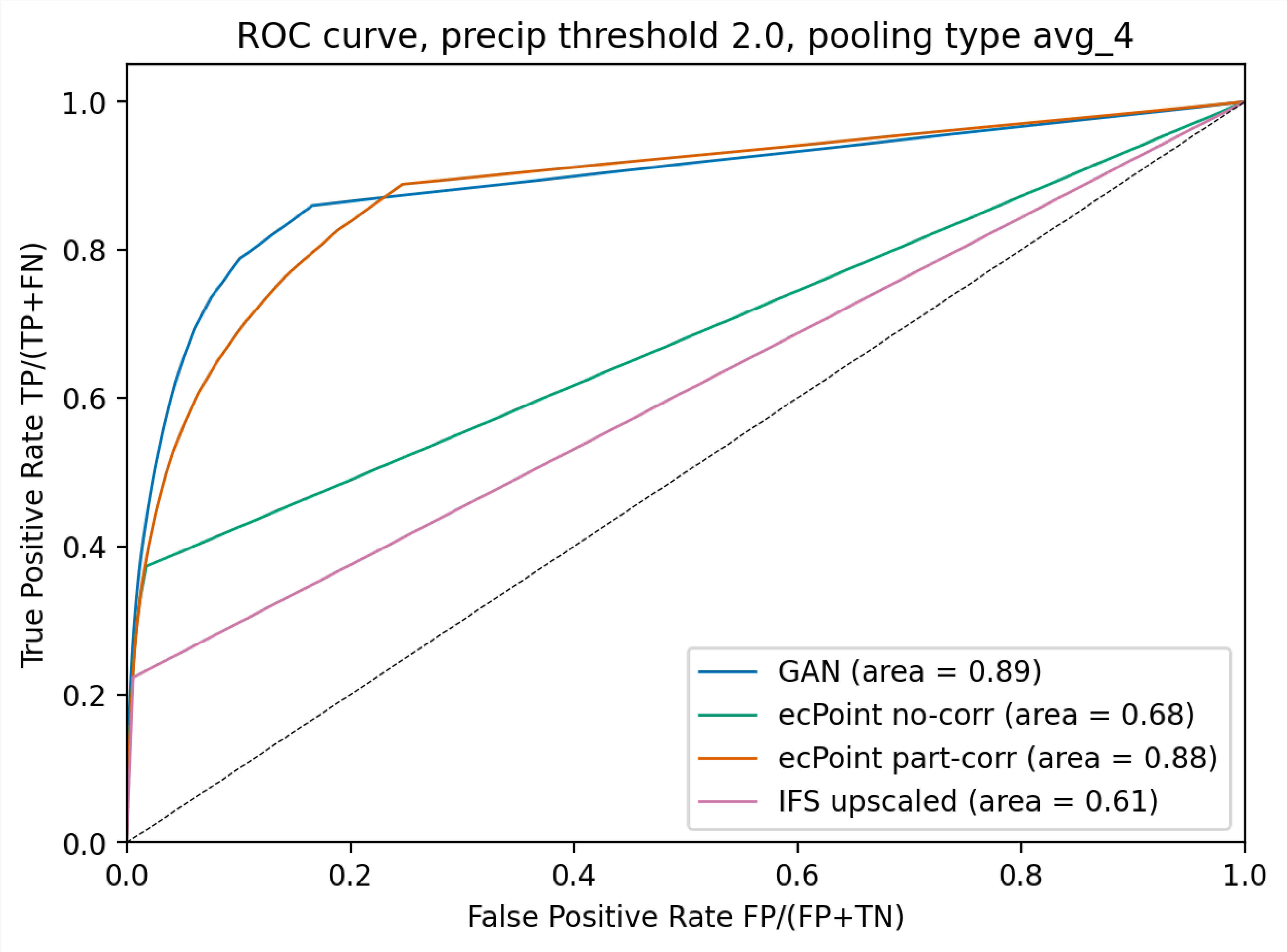


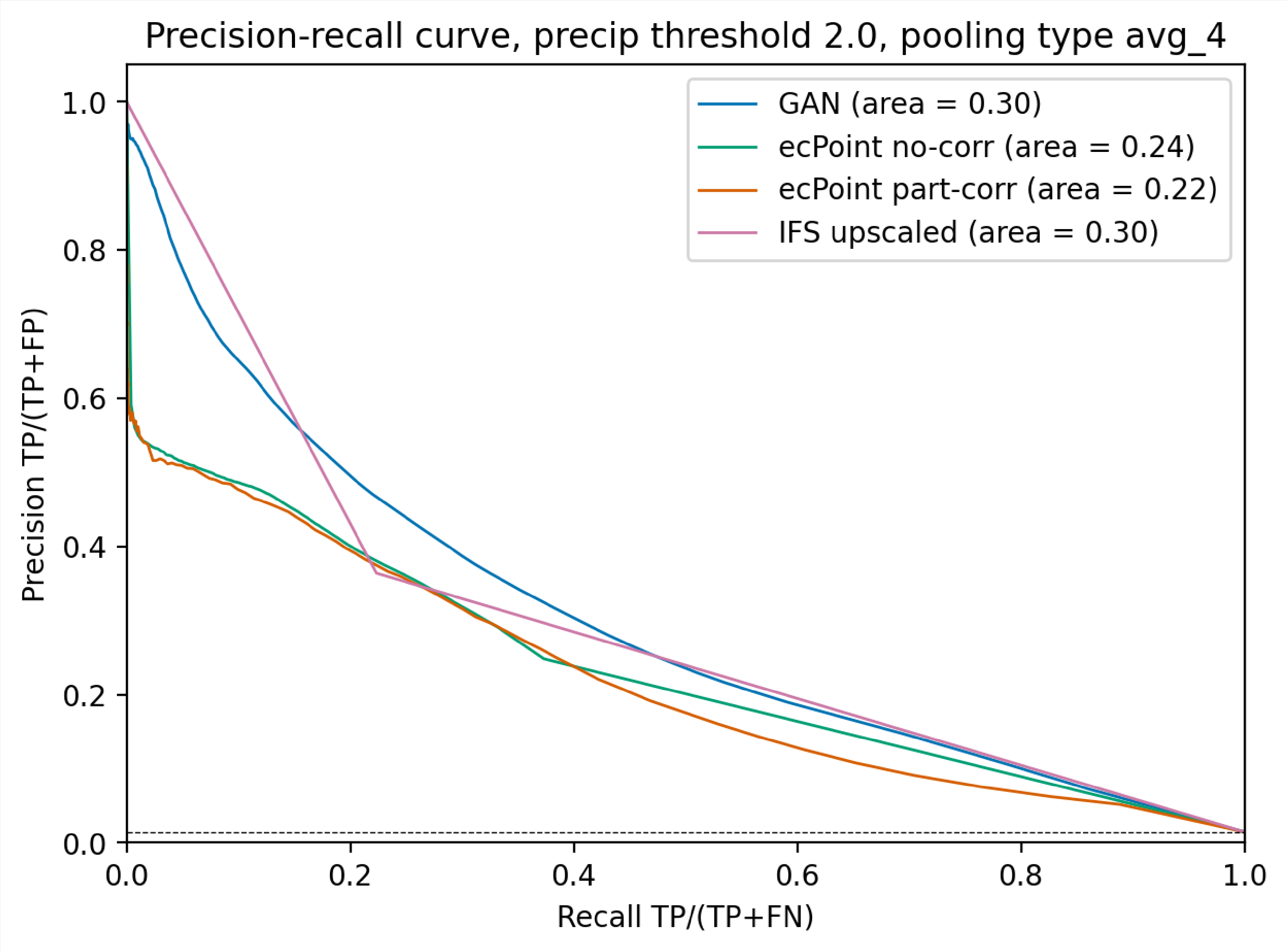
Solid = “ensemble skill”

Dotted = “individual prediction skill”

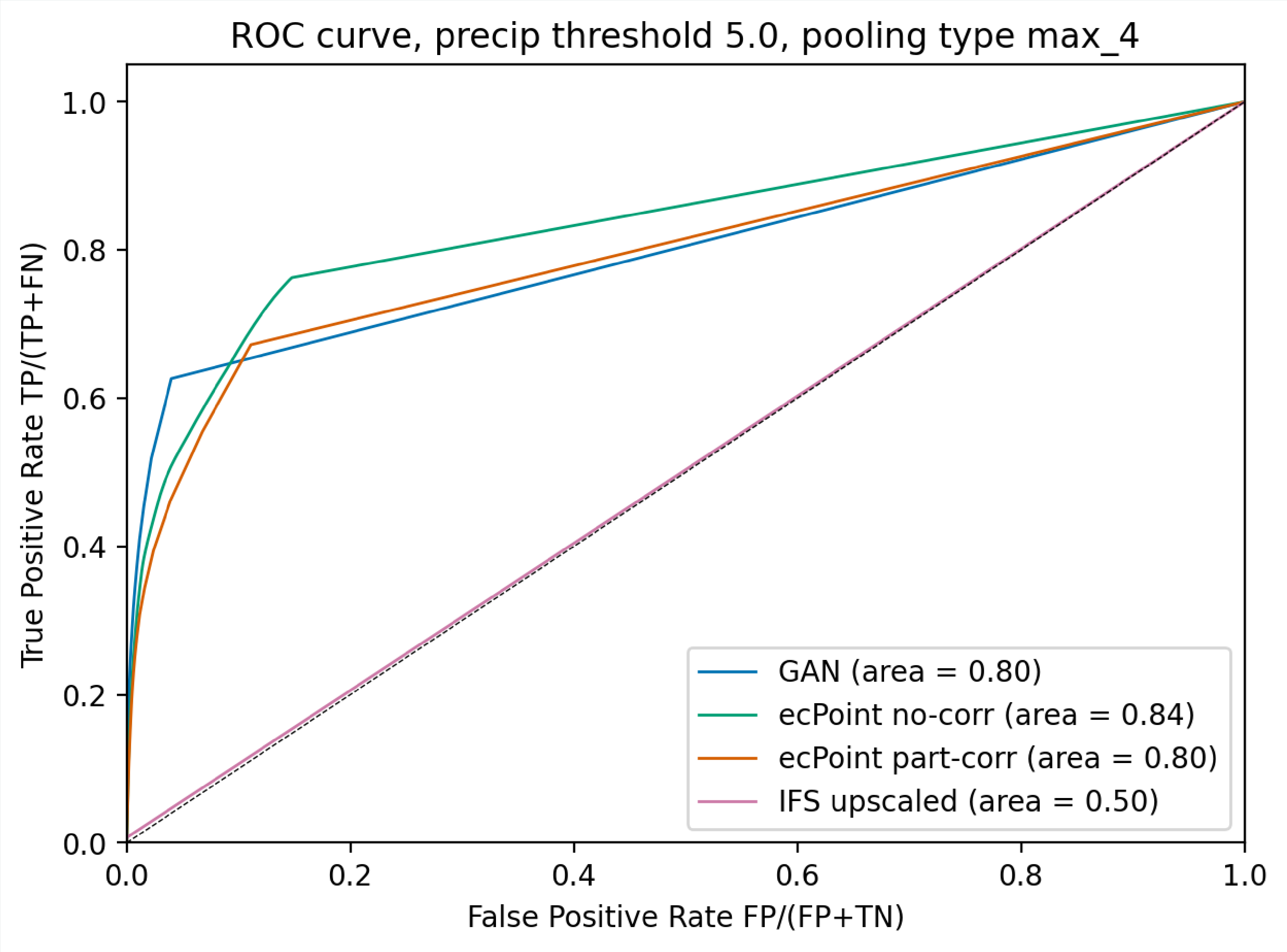


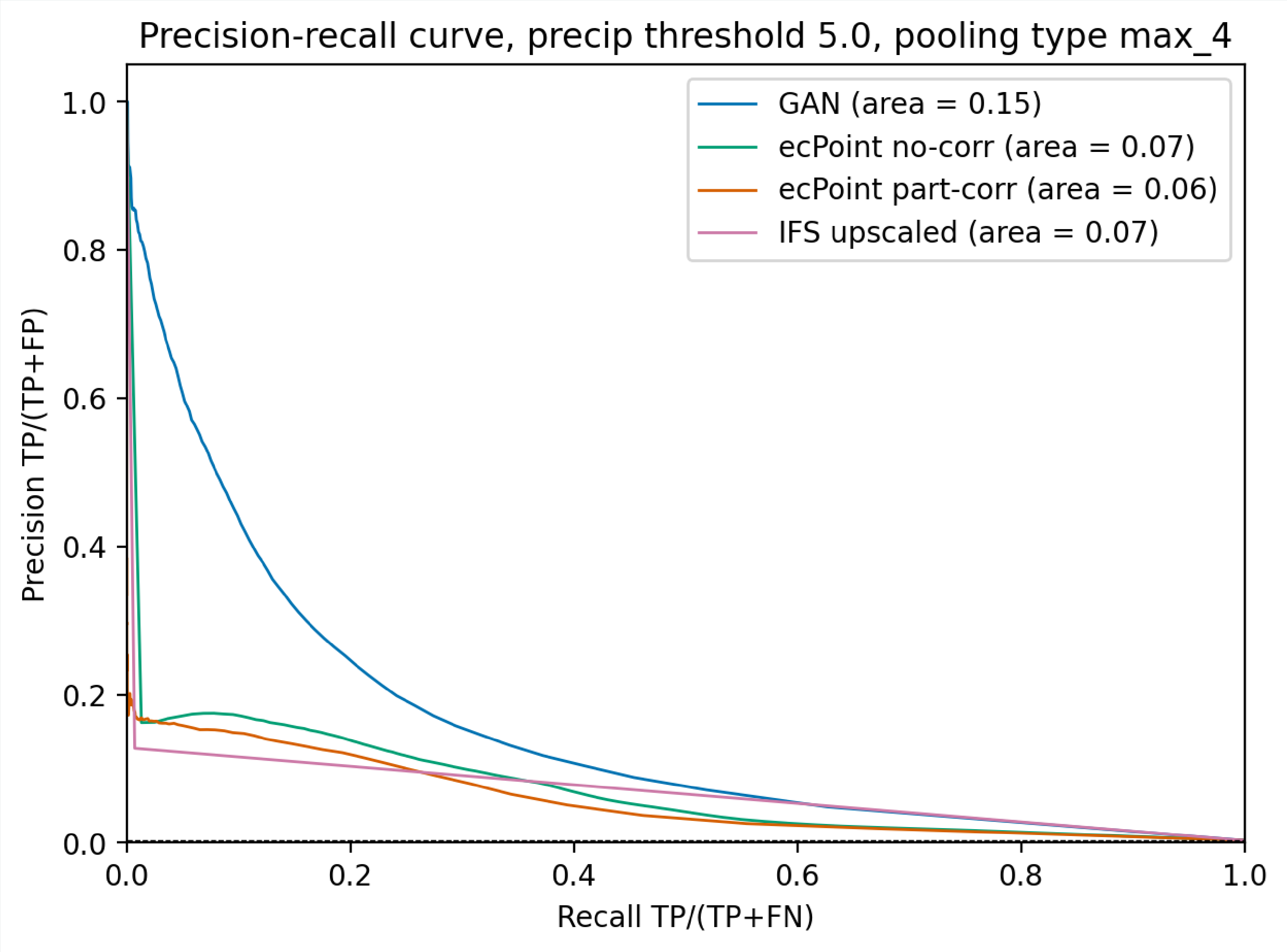
Solid = “ensemble skill”
Dotted = “individual prediction skill”

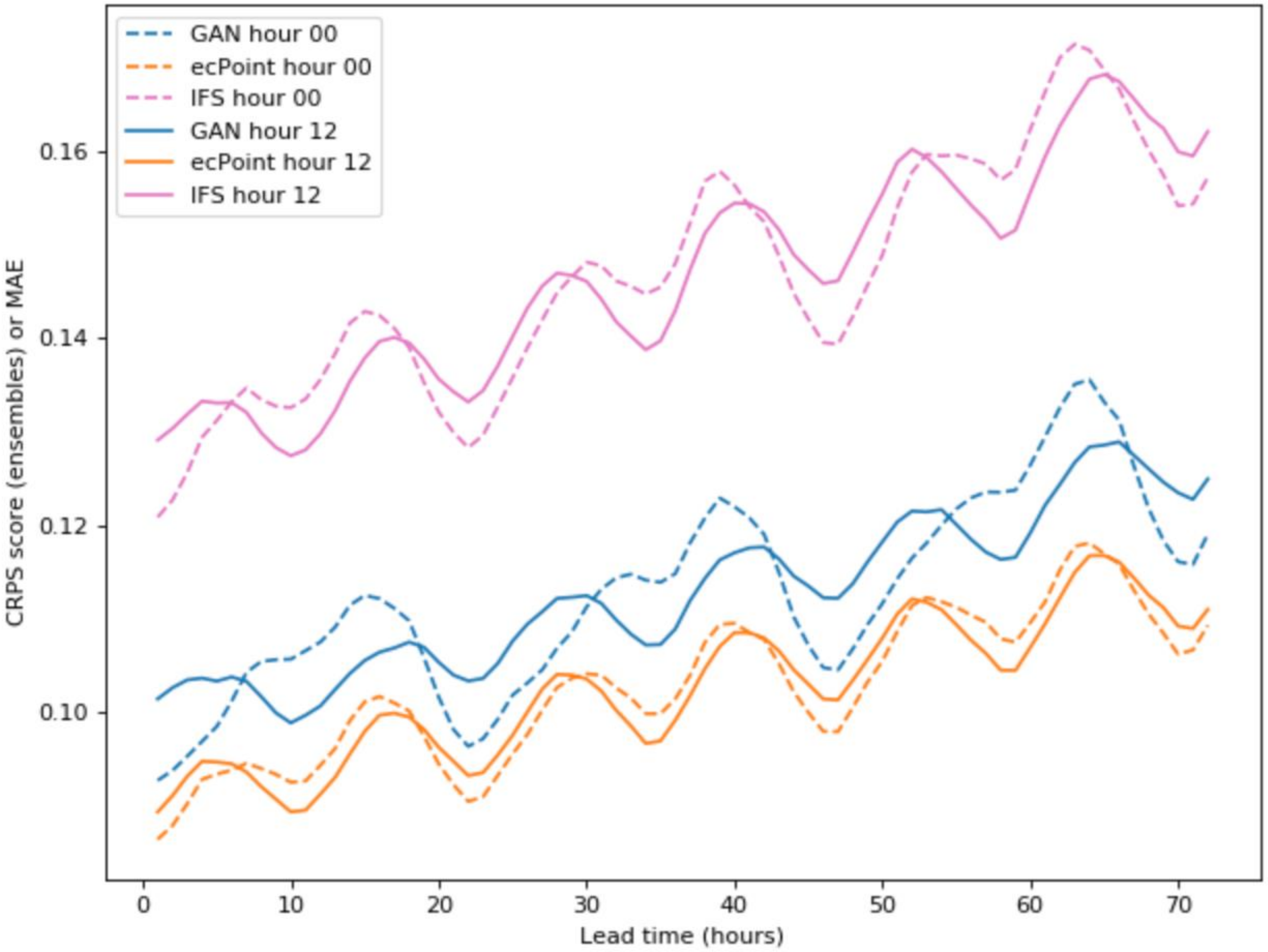

ROC curve, precip threshold 2.0, pooling type avg_4
True Positive Rate TP/(TP+FN)
False Positive Rate FP/(FP+TN)
0.0
0.2
0.4
0.6
0.8
1.0
GAN (area = 0.89)
ecPoint no-corr (area = 0.68)
ecPoint part-corr (area = 0.88)
IFS upscaled (area = 0.61)



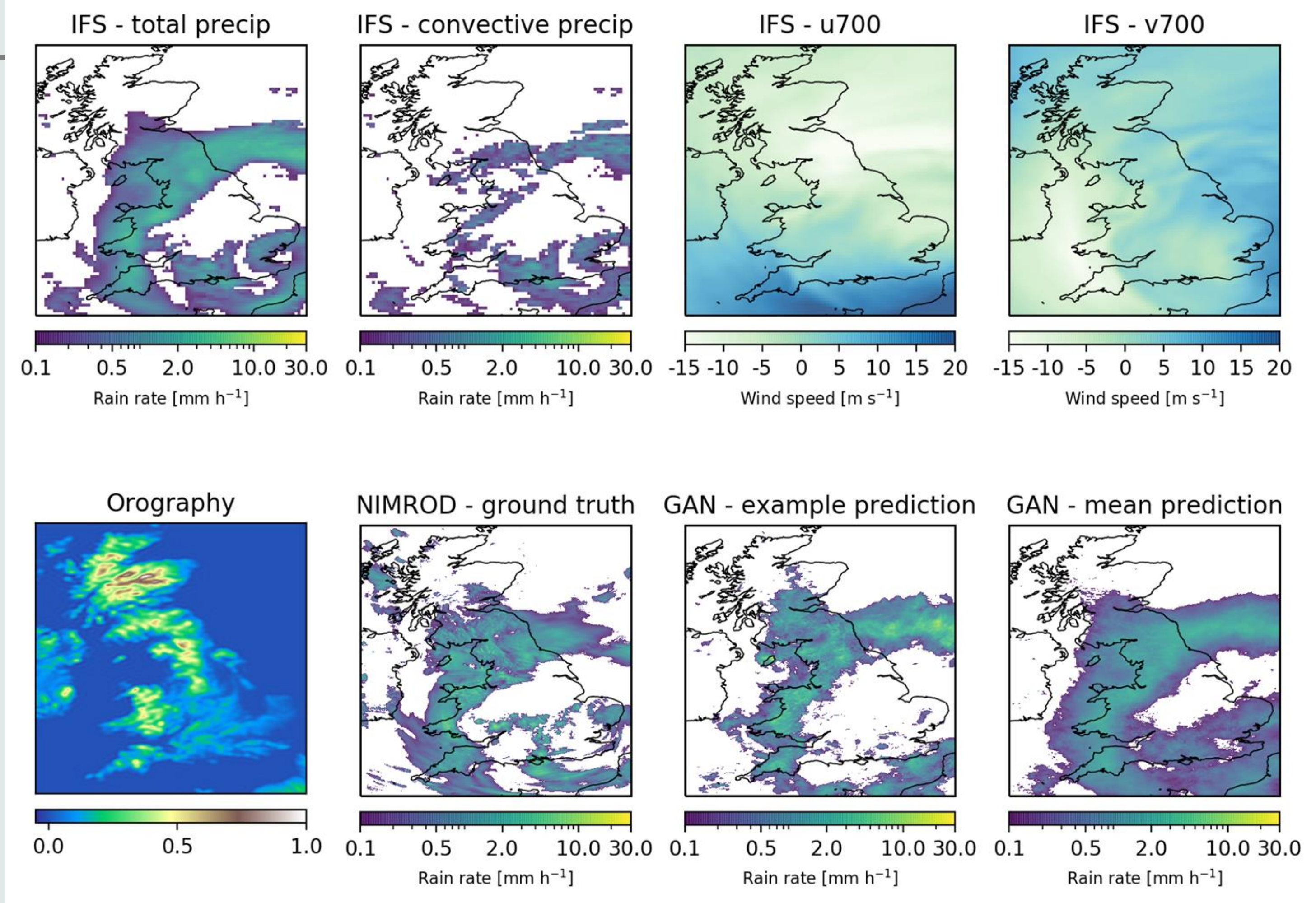
Precision-recall curve, precip threshold 2.0, pooling type avg_4
GAN (area = 0.30)
ecPoint no-corr (area = 0.24)
ecPoint part-corr (area = 0.22)
IFS upscaled (area = 0.30)
Precision TP/(TP+FP)
Recall TP/(TP+FN)
0.0
0.2
0.4
0.6
0.8
1.0


ROC curve, precip threshold 5.0, pooling type max_4
True Positive Rate TP/(TP+FN)
False Positive Rate FP/(FP+TN)
0.0
0.2
0.4
0.6
0.8
1.0
GAN (area = 0.80)
ecPoint no-corr (area = 0.84)
ecPoint part-corr (area = 0.80)
IFS upscaled (area = 0.50)


Precision-recall curve, precip threshold 5.0, pooling type max_4
GAN (area = 0.15)
ecPoint no-corr (area = 0.07)
ecPoint part-corr (area = 0.06)
IFS upscaled (area = 0.07)
Precision TP/(TP+FP)
Recall TP/(TP+FN)
0.0
0.2
0.4
0.6
0.8
1.0


GAN hour 00
ecPoint hour 00
IFS hour 00
GAN hour 12
ecPoint hour 12
IFS hour 12
CRPS score (ensembles) or MAE
Lead time (hours)
0.10
0.12
0.14
0.16
0
10
20
30
40
50
60
70

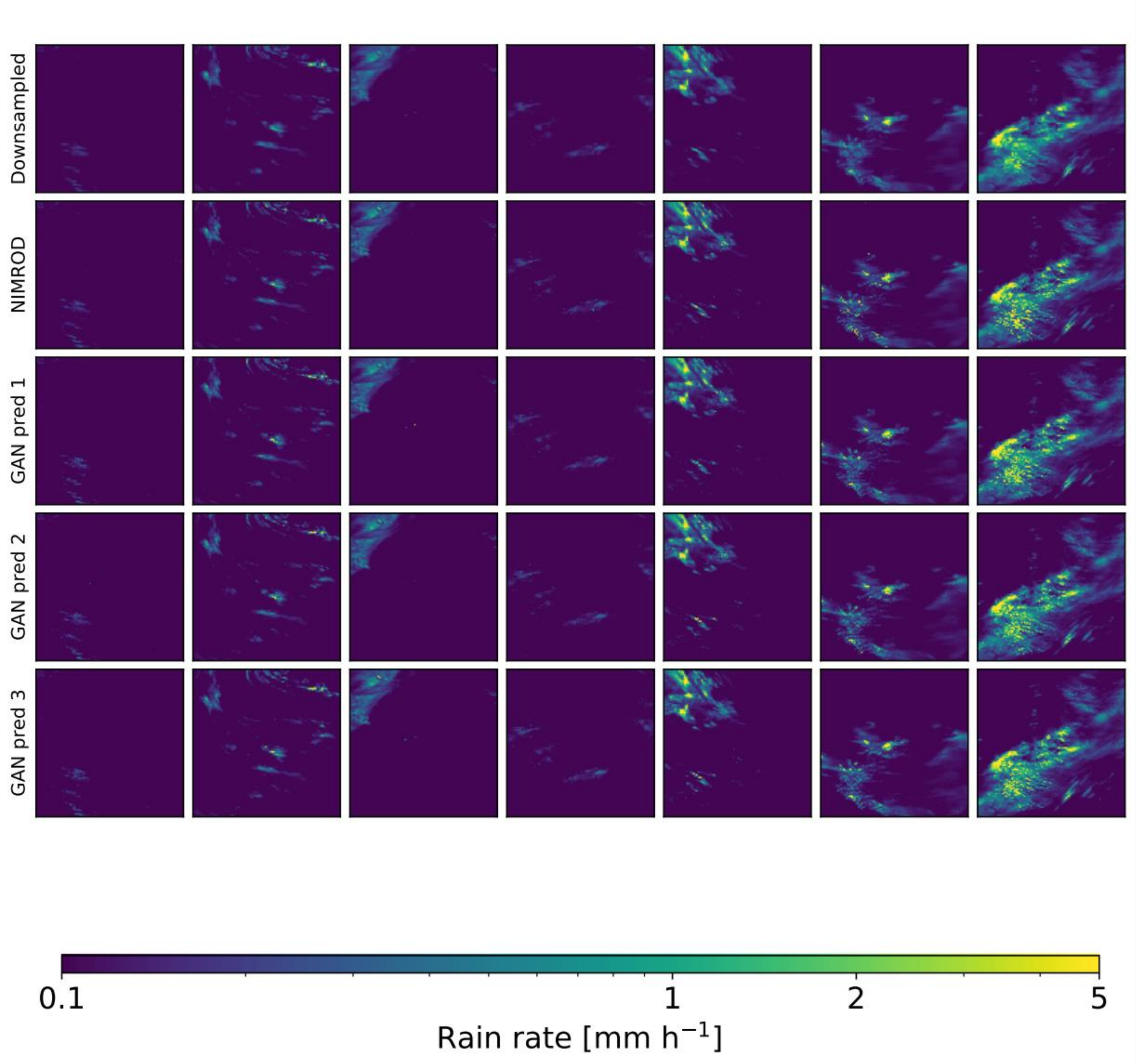
Improved model



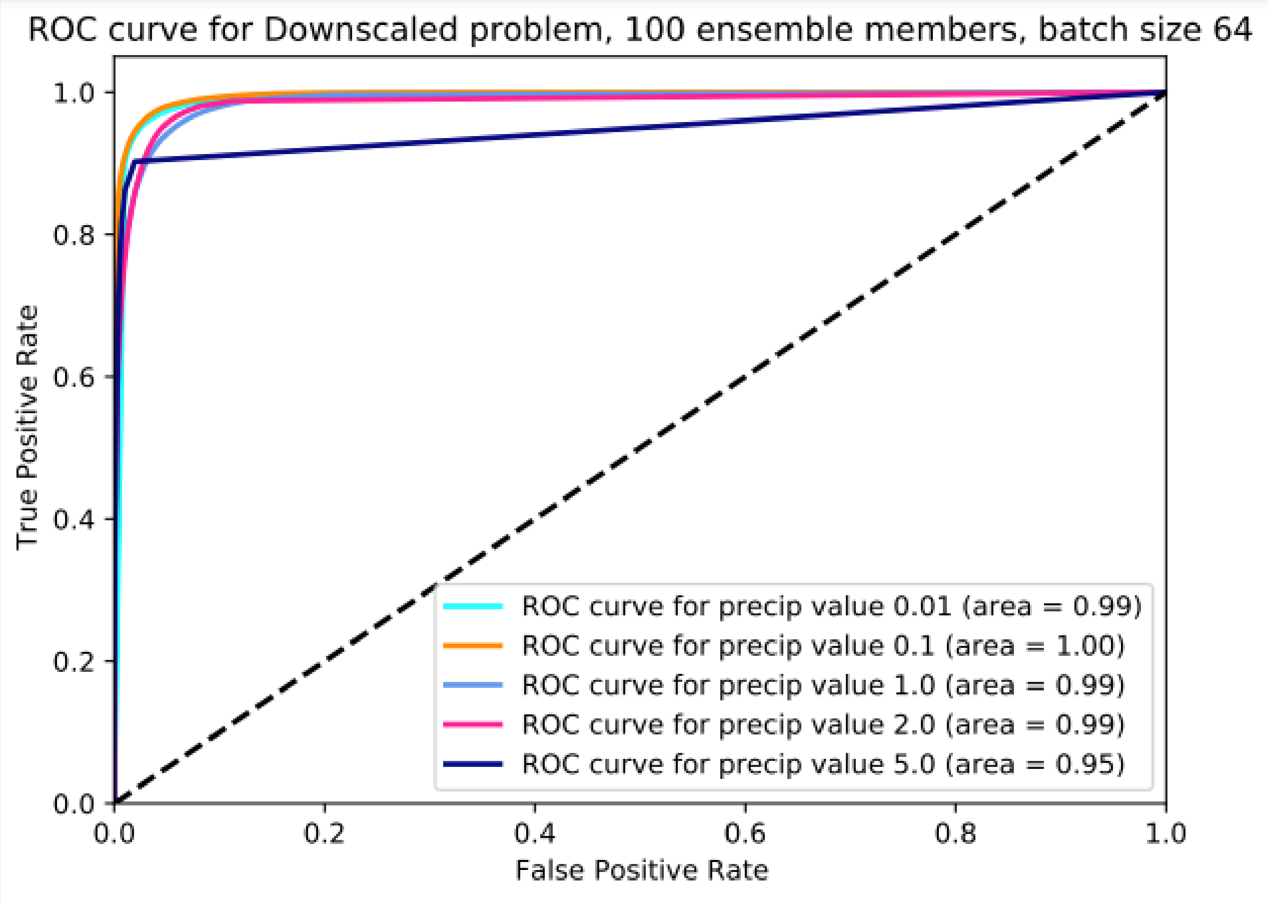
cGAN model

Downsampled probl

(c.f. Leinonen)



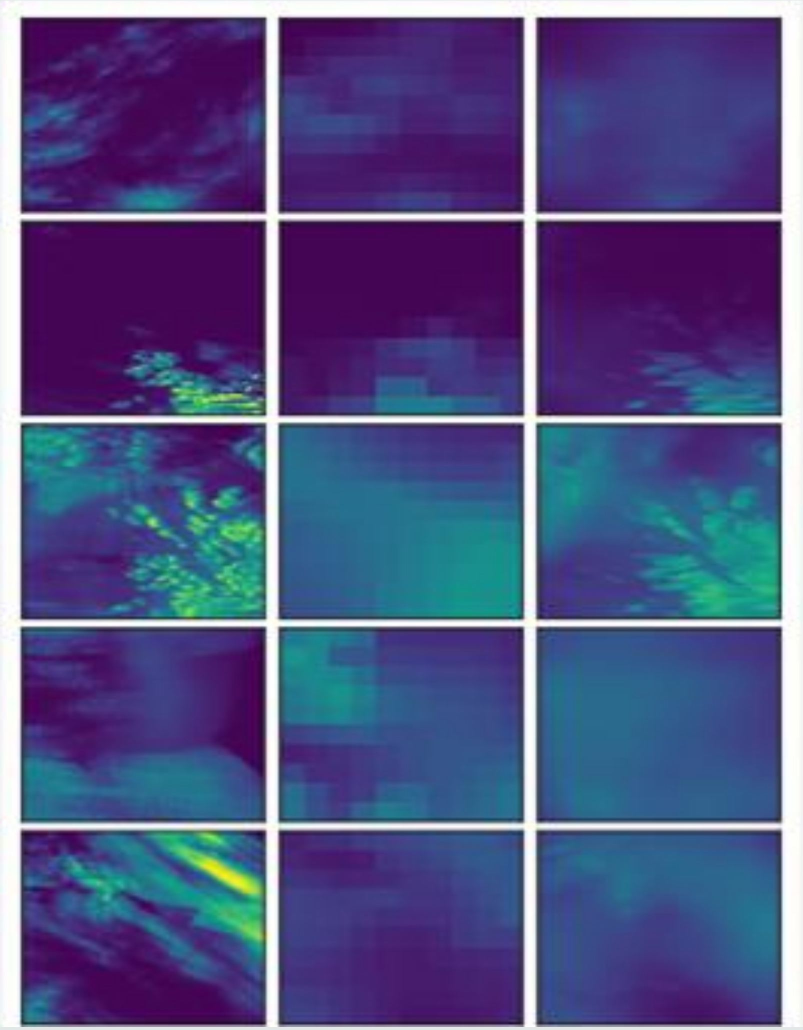
ROC curve - downscaled problem



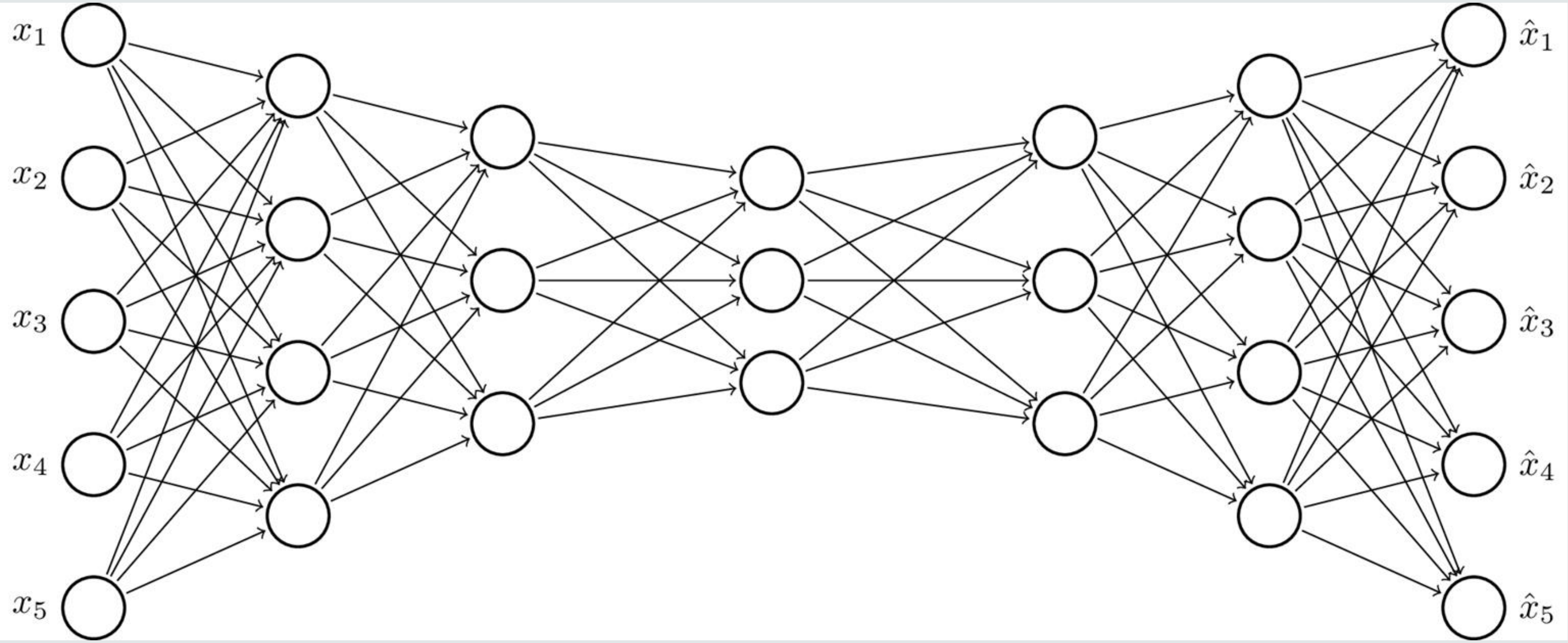
deterministic model

trainable parameters

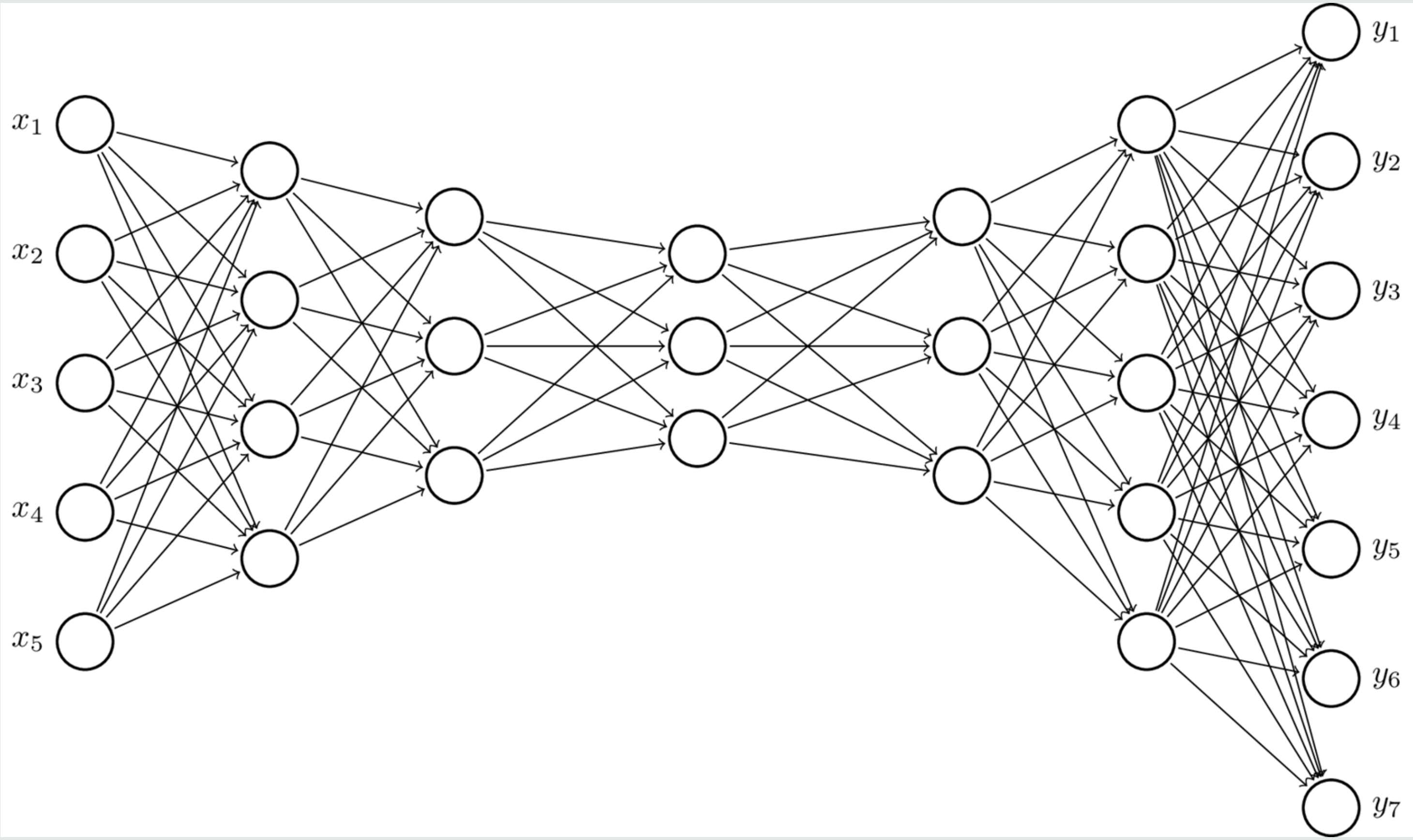
829,313



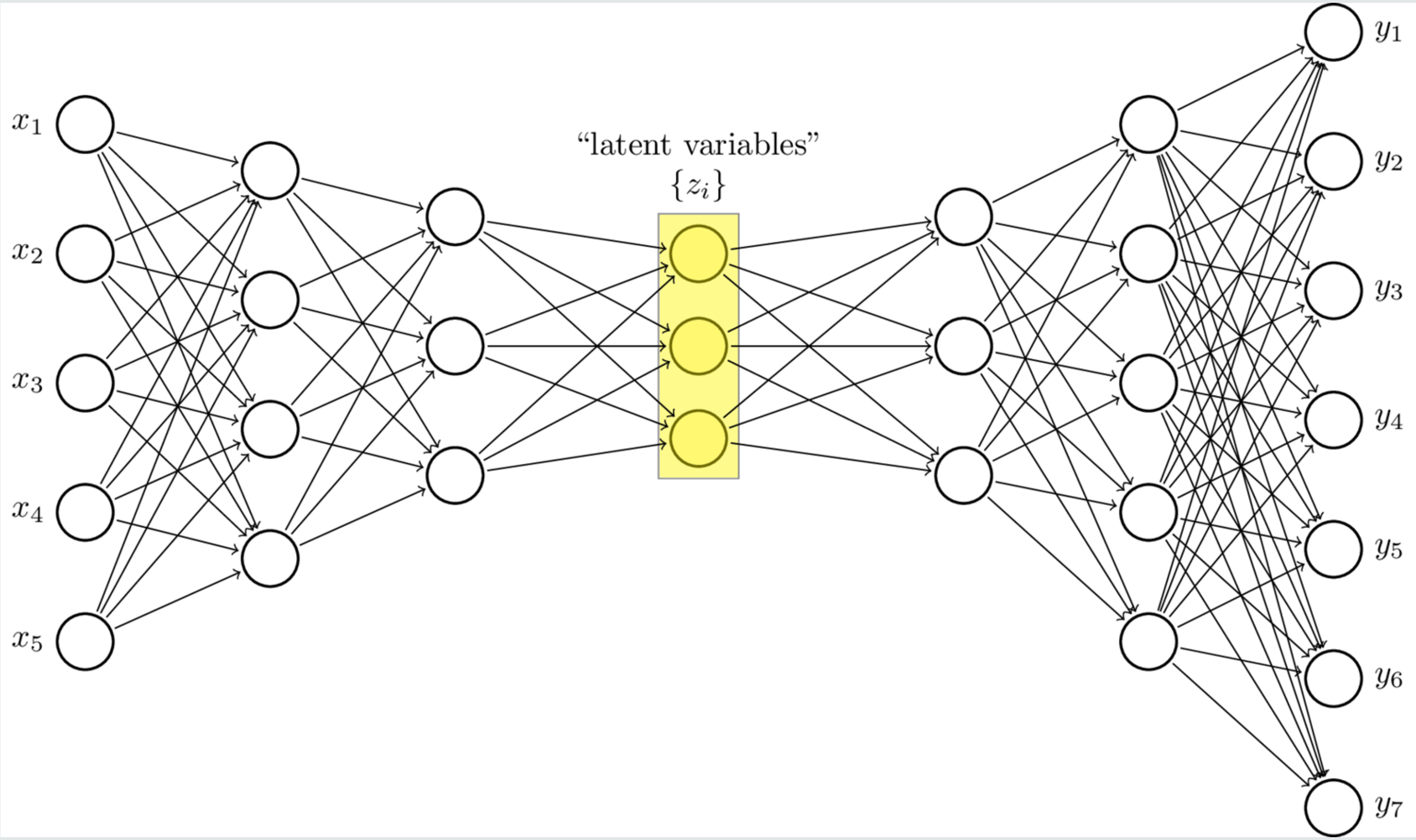
VAE intro - autoencoder



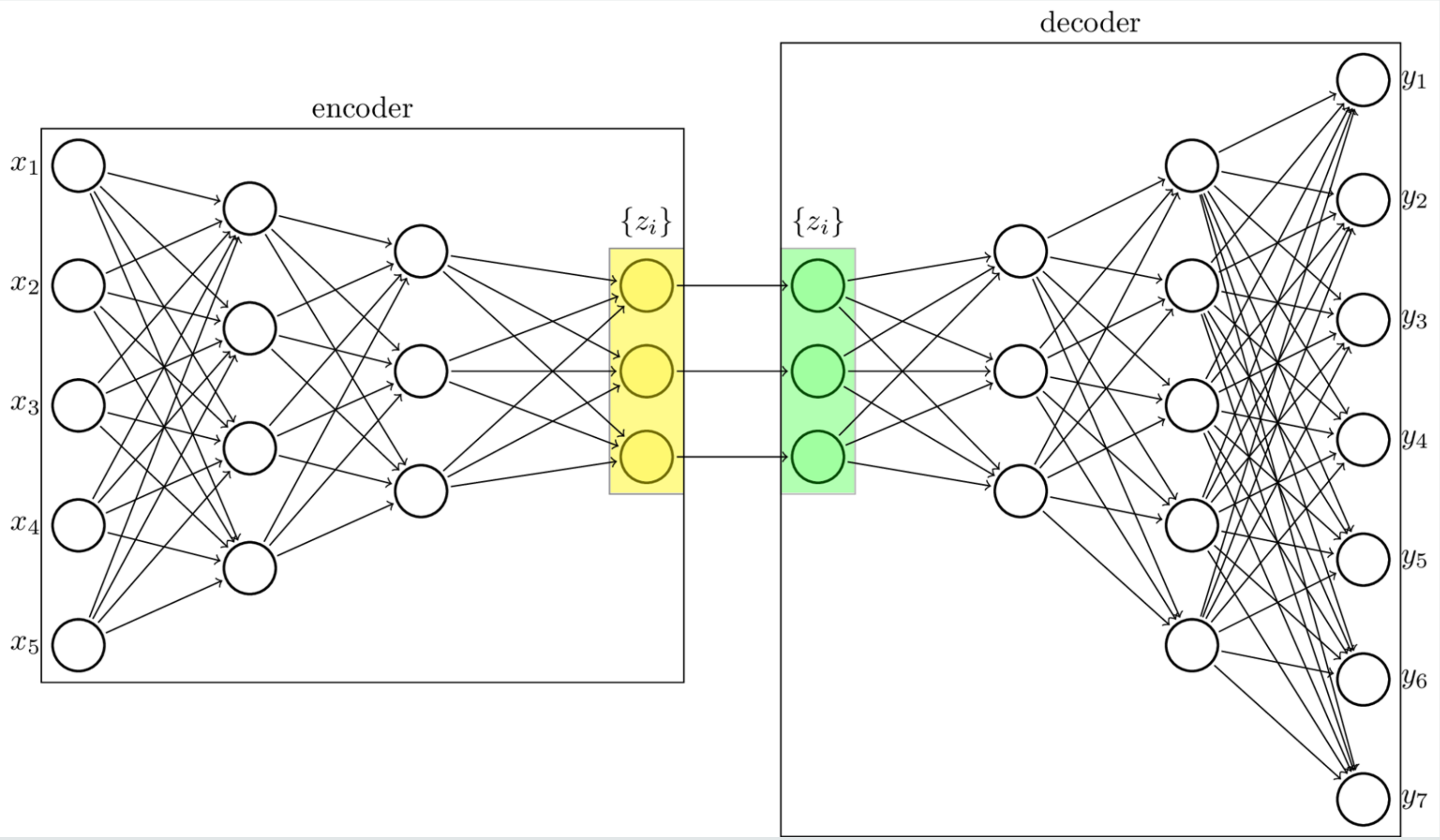
VAE intro - ~~auto~~encoder



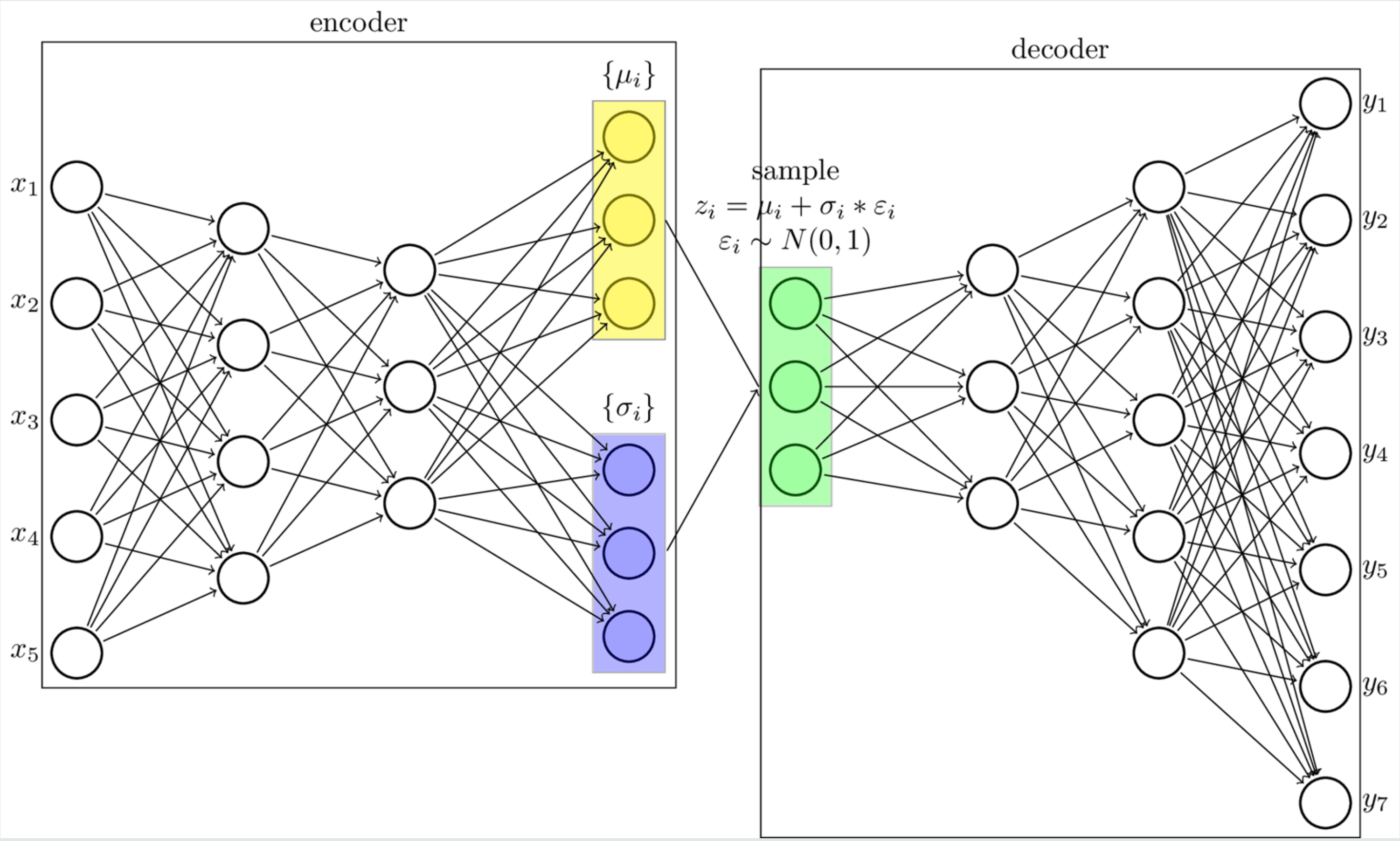
VAE intro - ~~auto~~encoder



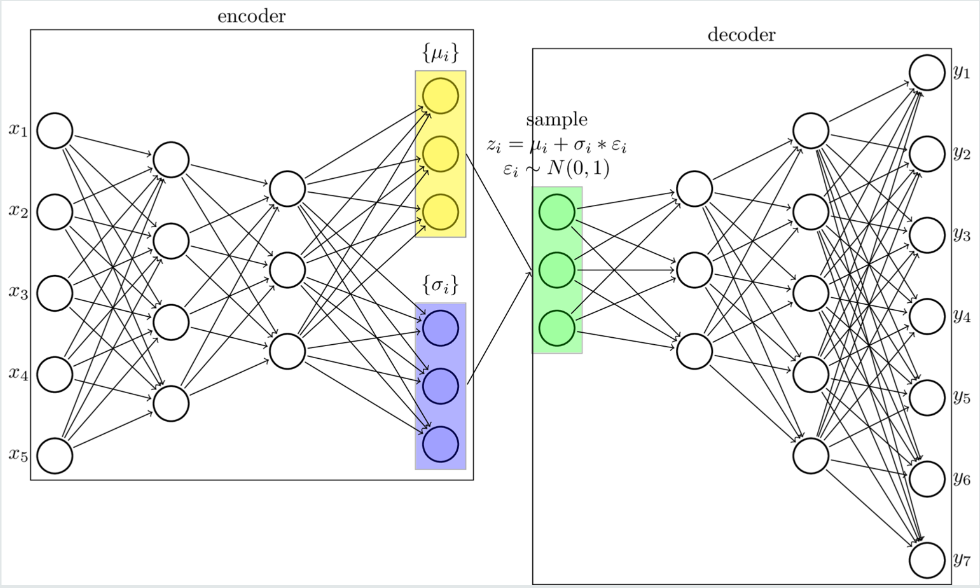
VAE intro - ~~auto~~encoder



VAE intro - (conditional) variational autoencoder



VAE intro - (conditional) variational autoencoder



Loss function also penalises latent variable distributions far from N(0, 1)

- Acts as regularisation
- Implemented as KL divergence

VAE loss function - mismatch term

Have explored traditional options including MSE, MAE.

Now using MSSSIM: Multi-Scale Structural Similarity Image Measure

- Based on “pixel-wise” dot product of images, applied at multiple scales
- Slightly better results than MSE, MAE, etc.
- Not a magic bullet, has given good results in other image generation problems; perhaps too ‘deterministic’ here.

Lack of suitable “mathematically expressible” loss function limits quality of VAE results

GAN

generative adversarial network

learns a loss that tries to classify an output as real or fake, while simultaneously training a generative model to minimise this loss function

a **minimax** game:

$$\mathcal{L} = \mathbb{E}_x\left[\log\left(\boldsymbol{D}(\boldsymbol{x})\right)\right] + \mathbb{E}_z\left[\log\left(1 - D(G(z))\right)\right]$$

the discriminator's estimate of the **probability** that **real data instance x is real**

GAN

generative adversarial network

learns a loss that tries to classify an output as real or fake, while simultaneously training a generative model to minimise this loss function

a **minimax** game:

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{x}}\left[\log\left(D(x)\right)\right] + \mathbb{E}_z\left[\log\left(1 - D(G(z))\right)\right]$$

the **expected value** over all **real data** instances

GAN
generative adversarial network

learns a loss that tries to classify an output as real or fake, while simultaneously training a generative model to minimise this loss function

a **minimax** game:

$$\mathcal{L} = \mathbb{E}_x\big[\log\big(D(x)\big)\big] + \mathbb{E}_z\big[\log\big(1 - D(\boldsymbol{G}(\boldsymbol{z}))\big)\big]$$

output of generator
when given input
random noise *z*

GAN

generative adversarial network

learns a loss that tries to classify an output as real or fake, while simultaneously training a generative model to minimise this loss function

a **minimax** game:

$$\mathcal{L} = \mathbb{E}_x\left[\log\left(D(x)\right)\right] + \mathbb{E}_z\left[\log\left(1 - \boldsymbol{D(G(z))}\right)\right]$$

discriminator's estimate of the **probability** that a generated, **fake instance is real.**

GAN
generative adversarial network

learns a loss that tries to classify an output as real or fake, while simultaneously training a generative model to minimise this loss function

a **minimax** game:

$$\mathcal{L} = \mathbb{E}_x\left[\log\left(D(x)\right)\right] + \mathbb{E}_z\left[\log\left(1 - D(G(z))\right)\right]$$

the **expected value** over all
random inputs to the **generator**

GAN

generative adversarial network

learns a loss that tries to classify an output as real or fake, while simultaneously training a generative model to minimise this loss function

a **minimax** game:

$$\mathcal{L} = \underbrace{\mathbb{E}_x\left[\log\left(D(x)\right)\right]}_{\text{refers to real data instances}} + \mathbb{E}_z\left[\log\left(1 - D(G(z))\right)\right]$$

refers to real data instances

GAN

generative adversarial network

learns a loss that tries to classify an output as real or fake, while simultaneously training a generative model to minimise this loss function

a **minimax** game:

$$\mathcal{L} = \mathbb{E}_x\left[\log\left(D(x)\right)\right] + \underbrace{\mathbb{E}_z\left[\log\left(1 - D(G(z))\right)\right]}_{\text{refers to fake (generated) data instances}}$$

refers to **fake (generated)** data instances

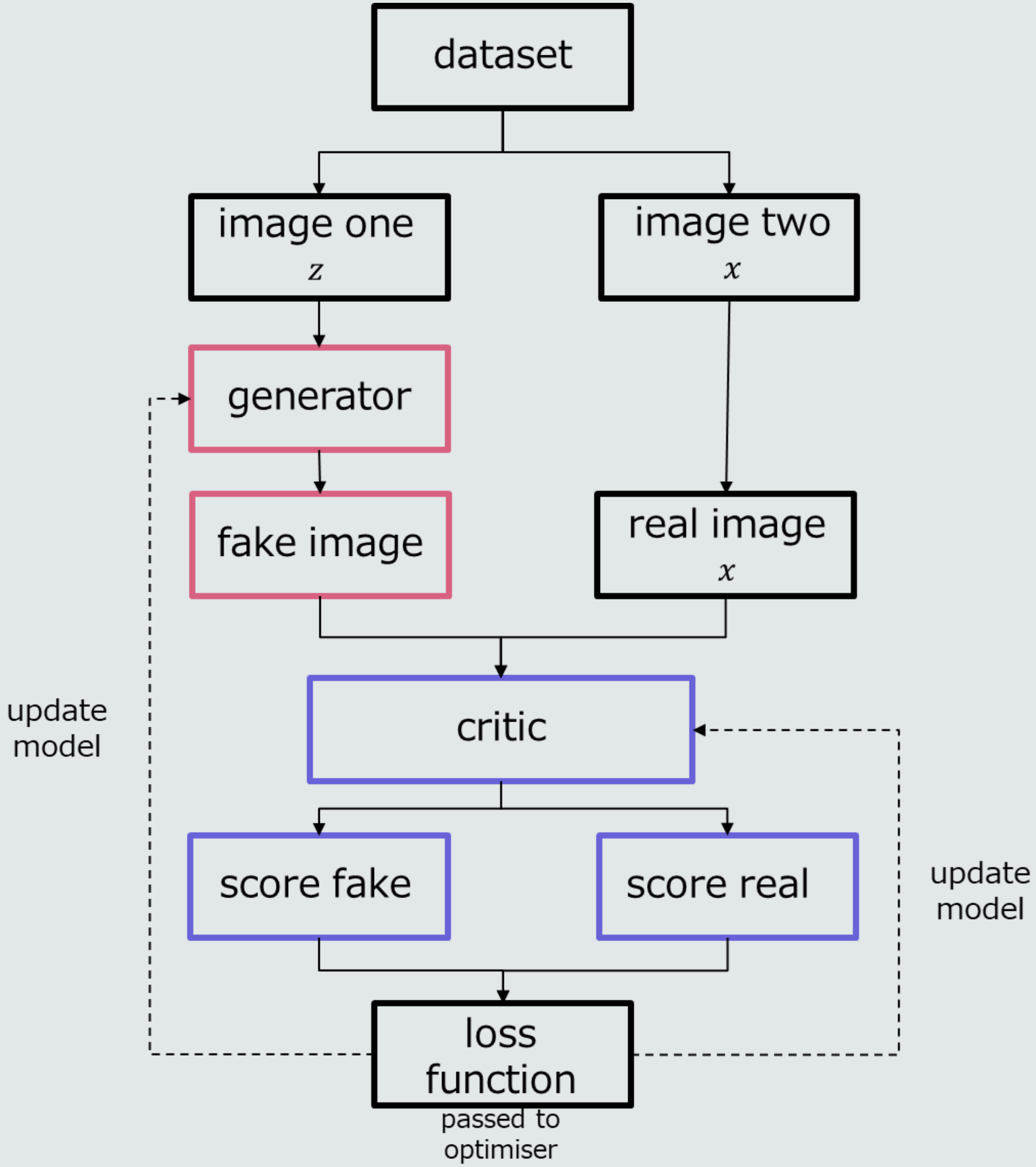
c-GAN

GAN loss:

$$\mathcal{L}_{GAN} = \mathbb{E}_x[\log(D(x))] + \mathbb{E}_z[\log(1 - D(G(x|z)))]$$

cGAN loss:

$$G^* = \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L_1}(G)$$



The RAPSD used in this study is defined as follows. The power spectrum of a 2-D image $f(x, y)$ of dimension $M \times N$ is defined as [33]

$$P(f) = |F(u, v)|^2 \tag{18}$$

where

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}. \tag{19}$$

Fig. 3 shows how spectral estimates $P(f)$ can be partitioned into annuli of width Δ for regular rectangular grids. Each annulus has a central radius f_r, a radial frequency, and $N_r(f_r)$ frequency samples. The sample mean of the frequency samples of $P(f)$ in the annulus $||f| - f_r| = \Delta/2$ about f_r is defined as the radially averaged power spectrum [31]

$$\overline{P}_r(f_r) = \frac{1}{N_r(f_r)} \sum_{i=1}^{N_r(f_r)} P(f_{r,i}) \tag{20}$$

where $f_r = [\sqrt{u^2 + v^2}]$ and $[\cdot]$ represents the nearest integer operator.

Wasserstein loss

based on the **earth-mover distance**

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_g}[f(x)]$$

$\mathbb{P}_r$ is the real data distribution, $\mathbb{P}_g$ is the generated model distribution

Wasserstein GAN

The loss functions themselves are simple:

Critic Loss: $\boldsymbol{D(x) - D(G(z))}$

The discriminator tries to maximise this function. In other words, it tries to maximise the difference between its output on real instances and its output on fake instances.

Generator Loss: $\boldsymbol{D(G(z))}$

The generator tries to maximize this function. In other words, It tries to maximise the discriminator's output for its fake instances.

WGAN

Wasserstein GAN

based on the **earth-mover distance**

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \le 1} \mathbb{E}_{x \sim \mathbb{P}_r}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_g}[f(x)]$$

$\mathbb{P}_r$ is the real data distribution, $\mathbb{P}_g$ is the generated model distribution

f is a **K-Lipschitz function**

$$|f(x_1) - f(x_2)| \le K|x_1 - x_2|$$

WGAN
Wasserstein GAN

based on the **earth-mover distance**

$$\mathcal{L} = \mathbb{E}_x[D(x)] + \mathbb{E}_z[D(G(z))]$$

$$D \in \mathcal{D}$$

$\mathcal{D}$ is the set of 1-Lipschitz functions

WGAN-GP

Wasserstein GAN with gradient penalty

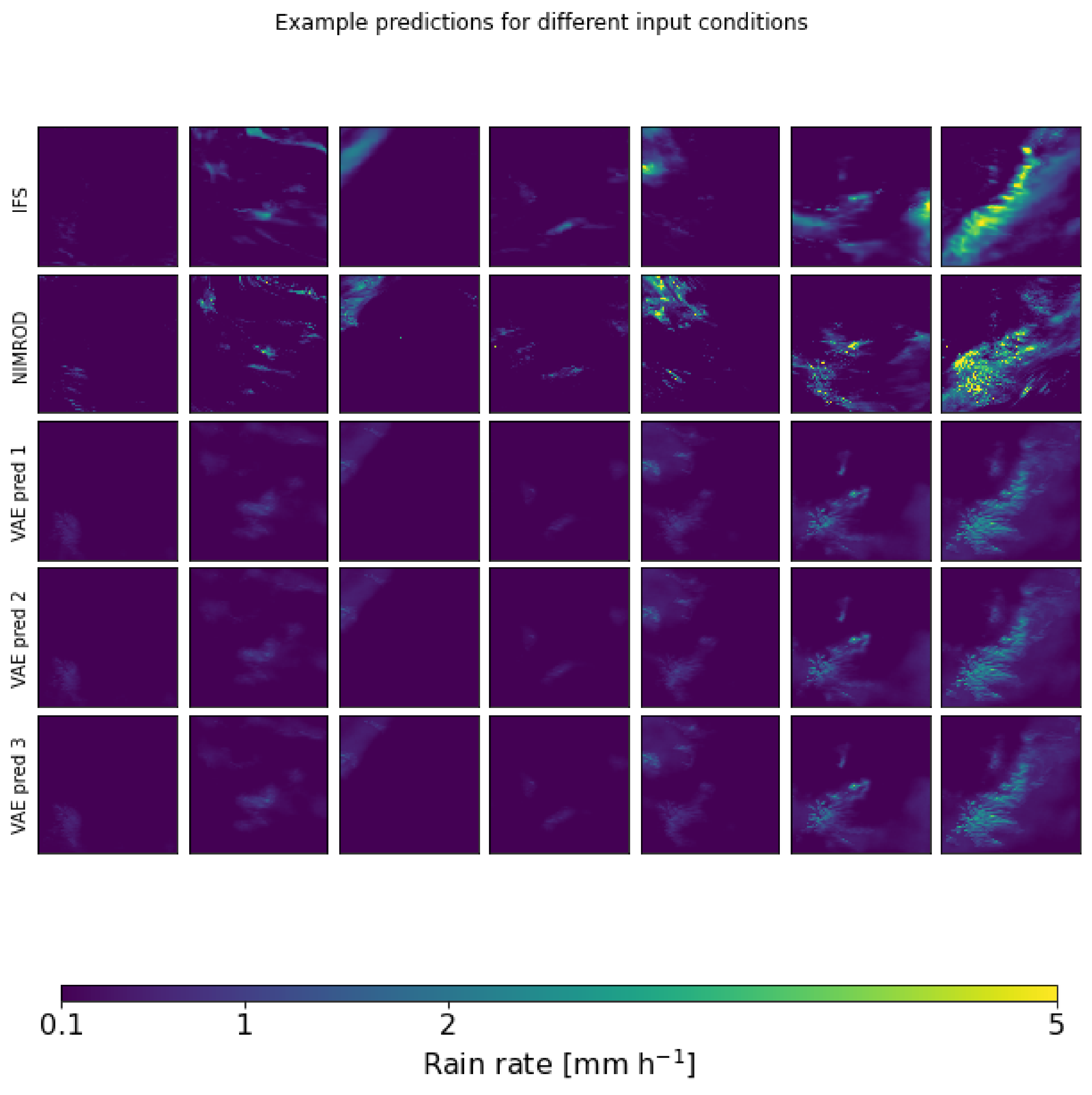
$$\mathcal{L} = \underbrace{E_x[D(x)] + E_z[D(G(z))]}_{\text{original critic loss}} + \underbrace{\lambda E_{\hat{\boldsymbol{x}}}[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2]}_{\text{gradient penalty}}$$

$\hat{\boldsymbol{x}}$ is a random sample from the probability distribution $\mathbb{P}_{\hat{\boldsymbol{x}}}$ which is implicitly defined by sampling uniformly along straight lines between a pair of points sampled from the real data distribution $\mathbb{P}_r$ and the generated distribution $\mathbb{P}_g$.

VAE model

full problem

- Too blurry
- Too little variation



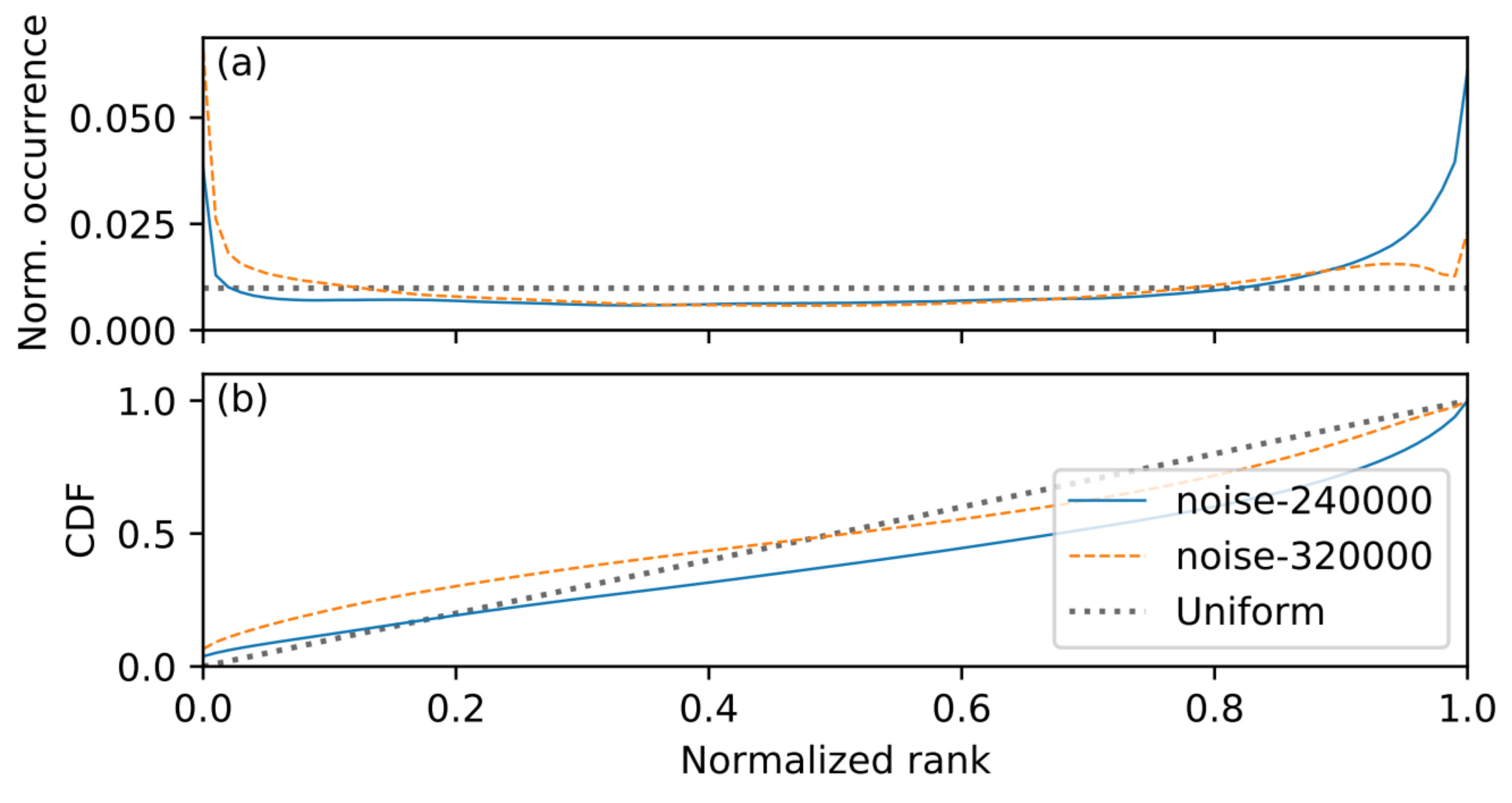
Rank histograms plot (cGAN)

$$r = \frac{N_s}{N_p}$$

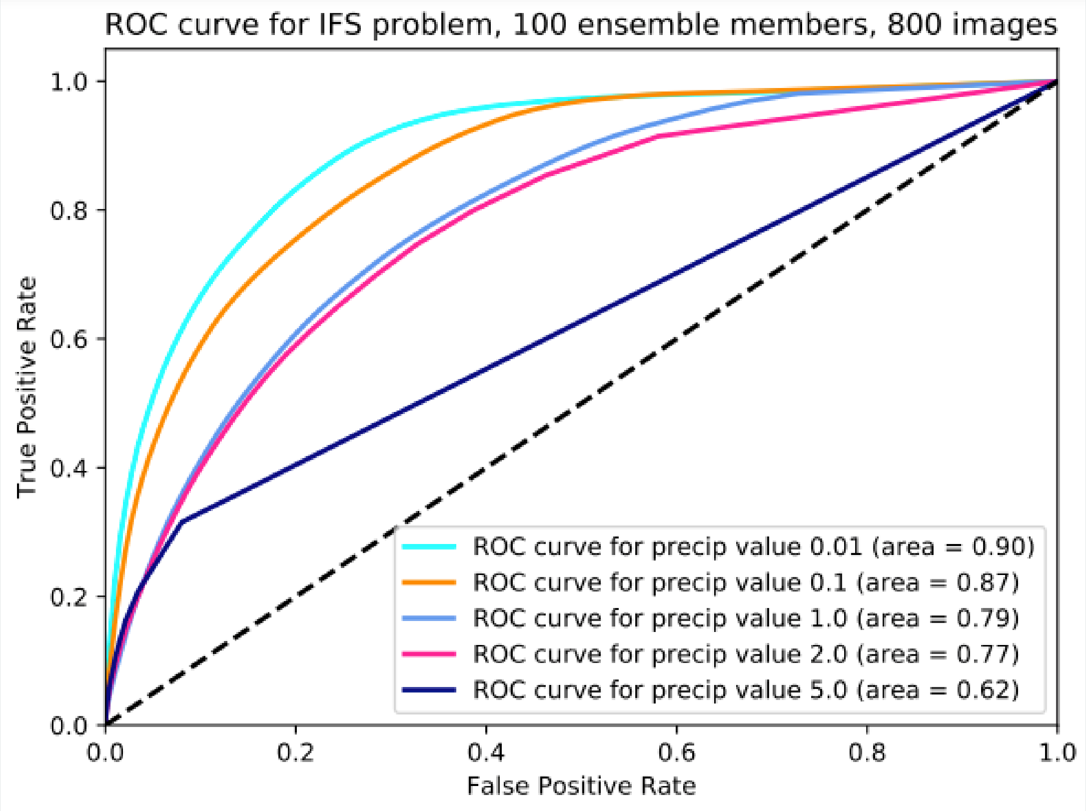
rank

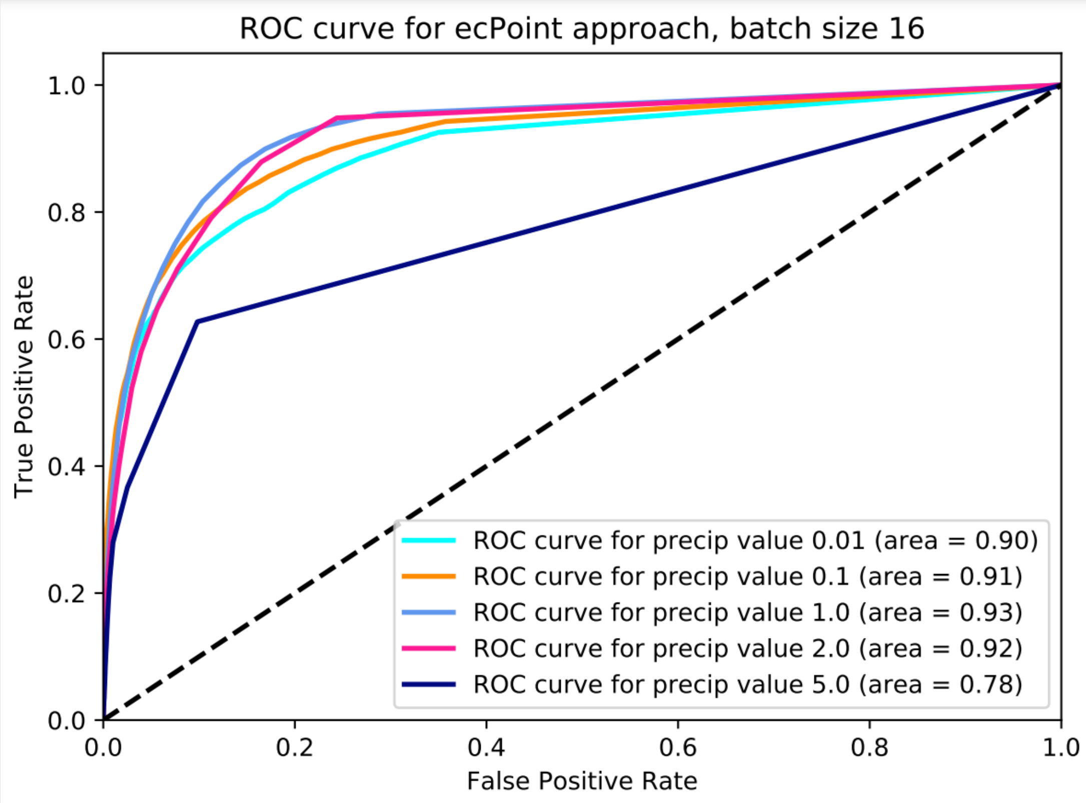
no. samples where pixel value is smaller than truth

total no. predictions



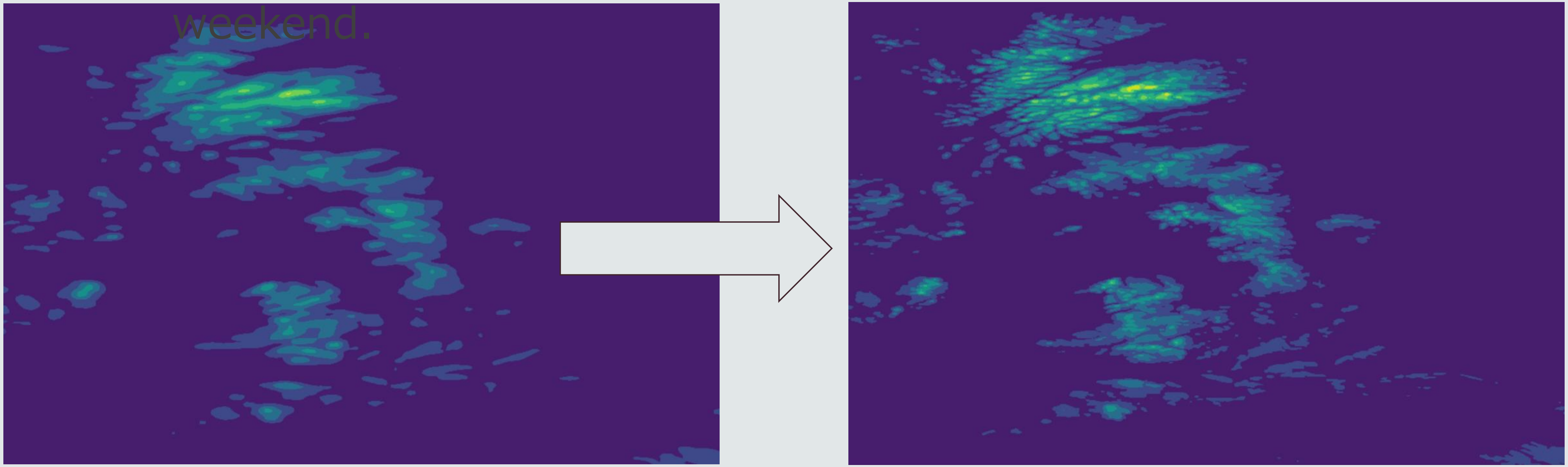
ROC curve


ROC curve for IFS problem, 100 ensemble members, 800 images
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve for precip value 0.01 (area = 0.90)
ROC curve for precip value 0.1 (area = 0.87)
ROC curve for precip value 1.0 (area = 0.79)
ROC curve for precip value 2.0 (area = 0.77)
ROC curve for precip value 5.0 (area = 0.62)


ROC curve for ecPoint approach, batch size 16
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve for precip value 0.01 (area = 0.90)
ROC curve for precip value 0.1 (area = 0.91)
ROC curve for precip value 1.0 (area = 0.93)
ROC curve for precip value 2.0 (area = 0.92)
ROC curve for precip value 5.0 (area = 0.78)

Improving orographic resolution

Previously we were training on ~4km orography. Have obtained 1km orography and will re-train on 1km this weekend.



Motivation - check if model produces e.g. rain shadowing (could not see much detail at 4km)

Tim Hewson feedback

So my initial impressions are these:

1. The GAN forecast realisations generally look quite reasonable / plausible, and better than I maybe expected, though I wasn't completely sure what to expect (!). However :
2. Sometimes there are patches of rain in areas where chances are *probably* so low that they should not be there (although I could have a better idea if you provided dates and times for all the cases)
3. The handling of moving convective cells leaves a lot to be desired - stripes of large totals should be the norm, with dry gaps inbetween, but you don't really see that at all (e.g. in the extremes case) - instead the picture is blurry.

Credibility of your GAN output in a forecaster environment would be hit somewhat by items 2 and 3, which would I'm sure be better handled by a LAM ensemble. Indeed it might be interesting for you to examine some LAM-EPS output of the same variable to see yourselves how it compares. By design ecPoint should do quite well with aspects 2 and 3, from a probabilistic perspective, even if it is not directly delivering high res totals plots like yours.