Supplementary Information for:

CEDDAR v1.0.2: Bridging physics and generative modelling for regional precipitation with controllable diffusion-based downscaling

Thea Quistgaard[1,2], Tanja Denager[2], Raphael J.M. Schneider[2], Jesper R. Christiansen[3], Simon Stisen[2], and Peter L. Langen[1]

[1]Department of Environmental Science, iClimate, Aarhus University, Roskilde, Denmark

[2]Department of Hydrology, Geological Survey of Denmark and Greenland, Copenhagen, Denmark

[3]Forest and Landscape Ecology, Department of Geoscience and Nature Management, Copenhagen University, Denmark

Contents

Overview

This Supplementary Information provides additional technical and methodological material complementary to the main manuscript. It includes extended derivations, full preprocessing documentation, expanded hyperparameter tables, architecture component study details, and extended evaluation metrics.

S1 Full Elucidated Diffusion Model (EDM) Formulation

This appendix provides a concise but complete theoretical background for the EDM formulation used in this work. The aim is to give the reader enough mathematical and conceptual background to reproduce the backbone of the model, without having thourough and detailed knowledge of the Karras et al. (2022) work or the broader diffusion-model literature. Our model follows the EDM parameterization of Karras et al. (2022), including explicit input, skip, output, and noise preconditioning coefficients, which stabilize training across a wide range of noise scales and ensure consistent denoising behaviour during inference

The key idea behind diffusion models comes from non-equilibrium thermodynamics (Sohl-Dickstein et al., 2015), and uses the general concept of diffusion: how a defined sample (image, distribution, fluid) goes through a diffusive process to end up as pure Gaussian noise. Diffusion models, as we know them from generative AI, focus on (1) *defining* and sampling a forward noising process (diffusion) of going sample-to-noise, and (2) *learning* a reverse (denoising) process. Figure 2 summarizes the EDM formulation used in this work, highlighting the continuous noise-level parameterization, σ-preconditioning during training, and the corresponding sampling procedure under conditioning.

S1.1 General diffusion framework

Diffusion models belong to a class of generative methods that generate samples by learning how to reverse a noise process that iteratively transforms data into pure Gaussian noise. The forward (*diffusion*) process is defined as an iterative increment of noise to a clean sample x_0, which in our work corresponds to the high-resolution precipitation field, which we denote as $x_0 \equiv x_{\mathrm{HR}}$. This means that the sample is perturbed by Gaussian noise of increasing magnitude, simplified to:

$$x_t = x_{\mathrm{HR}} + \sigma_t \epsilon \quad \epsilon \sim \mathcal{N}(0, I) \tag{1}$$

where x_{HR} is the clean data sample and σ_t is a monotonically increasing noise level. For sufficiently large σ_t, the sample becomes indistinguishable from isotropic Gaussian noise.

The reverse (*denoising*) process aims to reconstruct x_{HR} from x_t. This is modelled by a neural network - here a UNet, trained to estimate the score function, i.e. the gradient of the log density:

$$s_\theta(x_t, \sigma_t) \approx \nabla_{x_t} \log(p(x_t)). \tag{2}$$

This score estimate determines how to move our noisy sample x_t toward regions of higher probability density for the next iteration. A discretized form of the reverse process can be

written as:

$$x_{t-1} = x_t + g(\sigma_t) f_\theta(x_t, \sigma_t) + \sqrt{g(\sigma_t), \eta_t} \tag{3}$$

where f_θ denotes the learned noise-conditioned denoising direction (proportional to the score), $g(\sigma_t)$ is derived from the chosen diffusion SDE, and η_t denotes the stochastic sampling noise injected during sampling.

The training of a diffusion model aims to minimize the loss term that predicts the noise in a sample at time t:

$$\mathcal{L}_{\text{DM}} = \mathbb{E}_{x_{\text{HR}},\epsilon,\sigma} \left[\| \epsilon - f_\theta(x_{\text{HR}} + \sigma \cdot \epsilon, \sigma) \|^2 \right] \tag{4}$$

forcing the model to learn how noise at different - and any - levels manifest in the data space.

For climate modellers, diffusion models are attractive as they not only produce a single deterministic prediction, but can explicitly represent uncertainty and produce multiple physically consistent realizations given a single condition suite, a property that is central to uncertainty-aware downscaling.

S1.2 Conditional denoising

Formally, we seek to learn a conditional denoiser f_θ that maps noisy inputs and conditioning to a denoised estimate of the clean HR field, $\hat{x}_{\text{HR}}$,

$$f_\theta : (x_t, \sigma_t, c) \to \hat{x}_{\text{HR}}, \tag{5}$$

where x_t is the noisy version of x_{HR} at noise level σ_t, and c denotes the set of conditional inputs (the LR precipitation fields as well as other auxiliary predictors). We let x_{HR} denote the clean high-resolution precipitation field, sampled from the data distribution, directly corresponding to x_0 in standard diffusion notation.

S1.3 Elucidated Diffusion modification: training objective

Karras et al. (2022) introduced the Elucidated Diffusion Model which refines the vanilla diffusion framework described above to improve both numerical stability through stabilised gradients and sample quality. The most notable innovations of their work includes: (i) a noise prconditioning that standardizes data to $\text{Var}(x_{\text{HR}}) \approx 1$ and introduces a noise preconditioning that rescales both inputs and training targets, removing the sensitivity to the absolute scale of σ_t, (ii) a power-law noise schedule (compared to previous linear or cosine) $\sigma_t = (\sigma_{\max}^\rho + t \cdot (\sigma_{\min}^\rho - \sigma_{\max}^\rho))^{1/\rho}$ with tunable parameters $\sigma_{\min}$, $\sigma_{\max}$, ρ, and (iii) a training-time perturbation distribution $p(\sigma) \propto \exp\left(-\frac{(\log(\sigma) - P_{\text{mean}})^2}{2 \cdot P_{\text{std}}^2}\right)$ which ensures that all noise levels are sampled effectively. During the EDM training, each batch draws random noise σ-values from this log-normal distribution and then optimizes the loss:

$$\mathcal{L}_{\text{EDM}} = \mathbb{E}_{x_{\text{HR}},\epsilon,\sigma} \left[\| f_\theta(x_t, \sigma) - \hat{x}_{\text{HR}}(x_t, \sigma; x_{\text{HR}}) \|^2 \right], \tag{6}$$

where f_θ denotes the EDM denoiser and $\hat{x}_{\text{HR}}(x_t, \sigma; x_{\text{HR}})$ denotes the preconditioned training target derived from the clean HR field. This formulation yields smoother gradients, is less dependent on hyperparameter tuning, and gives faster convergence compared to the vanilla DM formulation.

S1.4 Residual learning: hypothesis and limitations

In statistical and deterministic downscaling, and in some Deep Learning image super-resolution frameworks, residual prediction is a natural hypothesis, as the high-resolution field or image is assumed to be a smooth perturbation of a coarse baseline. In our diffusion model, this corresponds to training the denoiser on the difference between HR and upsampled LR precipitation, i.e. replacing $\hat{x}_{\mathrm{HR}}$ with a preconditioned residual target:

$$\Delta x = x_{\mathrm{HR}} - x_{\mathrm{LR}^{\uparrow}}. \tag{7}$$

This effectively replaces the HR target in the EDM loss with the residual Δx in the denoising objective (up to standard EDM preconditioning):

$$\mathcal{L}_{\mathrm{EDM}} = \mathbb{E}_{x_{\mathrm{HR}},\epsilon,\sigma}\left[\left\|f_\theta(x_t,\sigma) - \Delta\hat{x}(x_t,\sigma)\right\|^2\right]. \tag{8}$$

Here $\Delta\hat{x}(x_t,\sigma)$ denotes the EDM preconditioned residual target.We omit standard EDM preconditioning constants for clarity. The residual formulation is applied consistently in the preconditioned target space.

While this formulation can improve numerical stability in under-conditioned settings - which we also observed in early development stages of our model - it implicitly assumes that the LR field provides a complete, meaningful and *structurally aligned* prior. However, when trying to learn the small-scale dynamics of precipitation fields, using the LR field as a direct prior in the loss can worsen performance, as the coarse-resolution LR is not a filtered version of the high-resolution truth but can differ distinctly in spatial organization, intermittency and event localization. As a result, the residual often contains the majority of high-entropy structure, instead of just a small correction term. This makes it difficult for the model to learn the complex large-to-small-scale relationships that *do* exist between LR and HR, just not in the form of a simple additive correction.

Our results from the structured architectural component study (main text Section 4.1 and Supplement Section S4) show that with a model that is already relatively stable (the B0 baseline model), the residual prediction ends up over-constraining the denoising process and suppresses stochastic expressiveness, leading to degraded extremes and spatial structure. For this reason, residual learning is not retained in the final model configuration.

S1.5 Dual-LR conditioning

We introduced early on a dual-LR conditioning option to explicitly expose systematic distributional differences, e.g. biases, between the low-resolution and high-resolution precipitation fields to the model. Specifically, we provided two conditioning channels derived from the same LR precipitation input, 1) one normalized using LR statistics (LR-in-LR space) and 2) one normalized using HR statistics (LR-in-HR space). Our hypothesis behind this conditioning scheme was that the normalization itself potentially could encode bias structures between the two datasets, allowing the network to directly infer how ERA5 deviates from DANRA under different scaling regimes. We thought this hypothesis especially interesting, since in a climate context, low- and high-resolution products are not simply related by spectral filtering, as is often the case in Computer Vision problems. LR and HR products differ in

intermittency, variance structure and extreme behaviour both due to physical parameterizations and resolution-dependent processes. By presenting the same LR field in two statistical "spaces", the model might in principle be able to learn an implicit bias-correction mapping embedded in the conditioning representation. We thought this relevant given two reasons: 1) classical statistical downscaling and bias-correction often rely on explicit distributional adjustments between driving and target datasets, and 2) when generalizing to future climate conditions, the statistical relationship between LR and HR responses may shift due to altered physics, convective regimes, or large-scale circulation. We hypothesised that a conditioning formula that explicitly encodes distributional differences might therefore provide a pathway to bias- or climate-aware generative downscaling. However, in our experiments (Supplement Section S4), dual-LR conditioning consistently degraded performance once the baseline model was properly calibrated. Rather than improving bias learning the additional conditioning channel appeared to introduce redundancy and increase optimization instabiity. This suggests that the diffusion backbone itself is already capable of learning the LR-HR relationship without encoding parallel statistics into it. Thus, we do not retain the dual-LR conditioning in the final configuration. But the broader idea of incorporating bias-aware conditioning remains of interest. Future work could investigate more structured approaches, such as explicit bias-correction residual branches, learnable distribution-alignment modules, or conditioning representations that adapt under climate-shift scenarios.

S1.6 Conditioning strategies

S1.6.1 Geographical conditioning and land-focused loss weighting

To emphasize the importance of accuracy on land while still training on the full land-sea domain, we apply a spatially varying loss weight derived from a coastline distance transform to emphasize predictive accuray over land and near-coastal regions. The weighting is derived from a coastline distance transform, assigning maximum importance to land pixels and gradually decreasing weights with increasing distance to shoreline. This design biases learning towards physically relevant small-scale precipitation structures over land without excluding ocean regions from training.

Let $m(i,j) \in \{0,1\}$ denote the land mask. We compute the ocean distance-to-land field $d_{\text{sea}}(i,j)$ using a Euclidean distance transform, which is zero on land and increase with ocean distance from the coastline. We then form a signed proxy field

$$s(i,j) = \alpha\, m(i,j) - d_{\mathbf{sea}}(i,j), \tag{9}$$

with $\alpha > 0$ (here $\alpha = 10$), and rescale it by min-max normalization

$$\bar{s}(i,j) = \frac{s(i,j) - s_{\min}}{s_{\max} - s_{\min}} \in [0,1]. \tag{10}$$

The per-pixel loss weight is defined as a bounded scale mapping

$$w(i,j) = w_{\min}^{\text{sea}} + \left(w_{\max}^{\text{land}} - w_{\min}^{\text{sea}}\right) S\big(\bar{s}(i,j)\big), \qquad S(x) = \frac{1}{1+\mathrm{e}^{-x}}. \tag{11}$$

Optionally, we normalize w per sample to preserve the overall loss scale, i.e. $\mathbb{E}_{i,j}[w(i,j)] \approx 1$. Note that the parameter α controls the relative separation between land and ocean distances and is chosen sufficiently large to ensure positive signed values over land prior to normalization.

S1.6.2 Seasonality: FiLM formulation

We incorporate seasonality through a Feature-wise Linear Modulation (Perez et al., 2018) conditioned on Day-Of-Year (DOY). The DOY input, d_t is first mapped to a continuous cyclic encoding, $\mathbf{y}_t = (\sin\theta_t, \cos\theta_t) \in \mathbb{R}^2$, to avoid steep transitions and discontinuities between December and January. In the EDM formulation, the noise level is also embedded through a small MultiLayer Perceptron (MLP), $c_{\text{noise}} = \frac{1}{2}\log(\sigma)$ and $\mathbf{t} = \text{MLP}_\sigma(c_{\text{noise}}) \in \mathbb{R}^D$, and the DOY encoding is mapped to the same dimension, $\mathbf{e}_{\text{doy}} = \text{MLP}_{\text{doy}}(\mathbf{y}_t) \in \mathbb{R}^D$, and added by $\mathbf{t}_{\text{comb}} = \mathbf{t} + \mathbf{e}_{\text{doy}}$. At each resolution layer in the U-Net, ℓ, with corresponding feature map $\mathbf{h}_\ell \in \mathbb{R}^{C_\ell \times H_\ell \times W_\ell}$, the FiLM parameters are determined from the combined conditioning vector $\mathbf{t}_{\text{comb}}$ and subsequently applied to the feature map $\mathbf{h}_\ell$ as:

$$(\boldsymbol{\gamma}_\ell, \boldsymbol{\beta}_\ell) = \text{Proj}_\ell(\mathbf{t}_{\text{comb}}), \qquad \text{FiLM}_\ell(\mathbf{h}_\ell) = (1 + \boldsymbol{\gamma}_\ell) \odot \mathbf{h}_\ell + \boldsymbol{\beta}_\ell. \tag{12}$$

S1.6.3 Auxiliary RainGate head

The RainGate head is a lightweight auxiliary network trained to predict per-pixel precipitation occurence. Pixels with a daily precipitation exceeding 0.1 mm day^{-1} are labeled as wet, and all others as dry. The task is formulated as a binary classification problem and optimized using a binary cross-entropy loss. The RainGate consists of a shallow convolutional network that takes as input the same large-scale dynamic meteorological and static geographical fields as the diffusion model. It outputs a single-channel logit field representing the probability of wet occurance at each pixel. The RainGate is trained jointly with the diffusion model, and gradients from the auxiliary loss propagate into the shared representation. The RainGate loss can optionally be used to also reweight the diffusion loss by pixel-wise reweighting. Importantly, in the final model configuration the RainGate loss is used only as an auxiliary supervision during training and does not perform any reweighting nor does it act as a hard gate during sampling. The RainGate's role is to encourage the shared feature representation to better separate wet and dry regimes, thus improving precipitation occurrence statistics without constraining stochastic variability.

S1.7 Full sampling schedule and σ^* integration

Sampling follows the EDM scheme from Karras et al. (2022) with modest churn.

To enable controllable scale-awareness, we apply a multiplicative noise scaling factor σ^* to the injected Gaussian noise:

$$x_{t-1} = x_t + g(\sigma_t)\, f_\theta(x_t, \sigma_t) + \sigma^*\, \eta_t, \tag{13}$$

where $g(\sigma_t f_\theta(x_t, \sigma_t)$ is the denoising term and $\sigma^* \eta_t$ is the injected Gaussian noise. This noise scaling factor only scales the injected Gaussian noise, and keeps the denoising drift

unchanged, thus avoids changing the learned score/denoiser. This means that σ^* can be interpreted as *how much stochasticity the denoiser has to "clean up" at each step.*

Formally, we let $\sigma_{i=0}^{N}$ be the standard EDM noise schedule from Karras et al. (2022) with $\sigma_0 = \sigma_{\max}$ and $\sigma_N = 0$. We then introduce a scale-aware inference factor σ^* that rescales the noise schedule at inference time only at late denoising steps (small noise). We define this late-ramp schedule through

$$i_0 = \lfloor r_{\text{start}}(N-1) \rfloor, \quad i_1 = \lfloor r_{\text{end}}(N-1) \rfloor \tag{14}$$

where i_0 and i_1 correspond to start and end indices of the ramp schedule, set from the user-defined fractions r_{start}, $r_{\text{end}} \in [0, 1]$. We then define a smooth transition weight function, $\text{Smoothstep}(u) = u^2(3 - 2u)$ to provide a continuous transition:

$$w_i = \begin{cases} 0, & i < i_0, \\ \text{Smoothstep}\left(\frac{i-i_0}{i_1-i_0}\right), & i_0 \le i \le i_1, \\ 1, & i > i_1. \end{cases} \tag{15}$$

From this weighting, we can then define our σ^*-modified noise schedule as:

$$\tilde{\sigma}_i = [1 + (\sigma^* - 1)w_i]\sigma_i, \qquad i = 0, ..., N \tag{16}$$

The Smoothstep function is a transition commonly used in computer graphics and numerical interpolation to avoid sharp discontinuities (Perlin, 1985). Note that σ^* rescales the *noise schedule*, rather than simply "adding more or less noise" to the final image.

The expected behaviour of changing σ^* can be separated in two cases:

1. $\sigma^* > 1$, where we inject more noise into the network. Here, the input to the network looks more corrupted, and the learned score field then responds by applying a stronger denoising drift. This drift will aggressively suppress high-frequency structure, and as a result, we go to an over-regularized regime, with smoother fields more coherent with the conservative Low Resolution condition, with less small-scale power, and collapsed variability.

2. $\sigma^* < 1$, where we inject less noise, and thus the network sees a cleaner input. The corresponding denoising drift is then weaker, and small-scale perturbations are not fully damped. This results in samples with more high-frequency power and pixel-scale variability.

This behaviour follows from the balance between stochastic forcing and denoising drift: reducing injected noise weakends the corrective action of the denoiser at late steps, allowing fine-scale fluctuations to persist. We further use a late ramp for σ^* which further enhances these effects. At late steps in the denoising process, σ_t is small, spatial structure is being finalized, and the denoiser controls correlation length and texture. In this regime, small changes in the injected noise have large relative effects on the Signal-to-Noise Ratio (SNR), and the balance between drift and noise is delicate.

This interpretation is consistent with diffusion theory and explains the observed monotonic changes seen in Supplement Section S5.

It is important to emphasize that σ^* modifies the stochastic sampling trajectory only at inference time. Because the network is trained on a fixed noise distribution, changing σ^* does not correspond to sampling from a different learned data distribution, but instead constitutes a controlled perturbation of the denoising path. Moderate deviations of σ^* thereby explore nearby regions of the actual learned manifold, while larger deviations push the sampler off-manifold, and out-of-distribution with respect to the training data.

This behaviour can be seen a an analogue to temperature scaling in other generative models (e.g. truncation or temperature scaling in GANs and diffusion models, (Karras et al., 2022; Brock et al., 2018): within a calibrated range, outputs remain physically plausible and interpretable, but beyond that range, samples may exhibit non-physical artefacts such as excess high-frequency power.

Scaling the stochastic term but not the drift maintains the EDM's calibrated denoising behavior while letting the user modulate the realized small-scale variability to some degree, in a calibrated area around the denoising path. Values of $\sigma^* > 1$ suppress high-frequency variance, creating smoother, more conservative fields coherent with LR condition, whereas $\sigma^* < 1$ increases fine-scale precipitation structure and ensemble spread. A late-ramp schedule limits the influence of σ^* to intermediate and low noise levels where spatial structure emerges. Because high-frequency structure emerges primarily during the late denoising steps (low σ_t), scaling the injected noise at those steps directly modulates the effective spatial correlation length of the output field.

S2 Data Preprocessing and Conditioning Pipeline

This section documents the complete preprocessing pipeline workflow required to set up data for training, validation and test datasets. All steps are implemented in the public data-processing scripts.

S2.1 Data sources

We use daily precipitation from two sources:

- DANRA v0.5: A high-resolution (2.5 km $\times$ 2.5 km) regional reanalysis covering Denmark for 1991-2020 (and extending), (Yang et al., 2022, 2021)
- ERA5: The corresponding global reanalysis (0.25° resolution) regriddid to the DANRA domain (Hersbach et al., 2020).

All fields are aligned to a fixed 180 $\times$ 180 grid covering Denmark and coastal surroundings (Figure 3).

S2.2 Preprocessing workflow, end-to-end

Figure S1 illustrates the complete pipeline. The main steps are:

1. *Regridding*: ERA5 precipitation is bilinearly regridded to the DANRA grid through the use of Climate Data Operators (CDO). This ensures consistent coordinate systems,

masks and land-sea structure. The data stored for this work is saved in this domain extend (589 × 789) defined by the DANRA bounds.

2. *Domain extraction*: A 180 × 180 area enclosing the Danish area and key surrounding ocean is extracted. All analysis rely on this domain.

3. *Cutout generation*: For training, 128 × 128 spatial patches are sampled randomly within ther 180 × 180 domain for each day. This increases sample variability and reduces memory load during training. Validation and test used deterministic, fixed 128 × 128 cutout locations (see all spatial figures in main text).

4. *Scaling*: precipitation exhibits long tails, zero-inflation and near-zero regimes, and to combat this in training, we apply a log-z-score transformation:

$$x' = \frac{log(x + \epsilon) - \mu}{\sigma}, \quad \epsilon = 0.01 \text{ mm day}^{-1} \tag{17}$$

   This ensure a close-to normal distribution for the model to learn from.

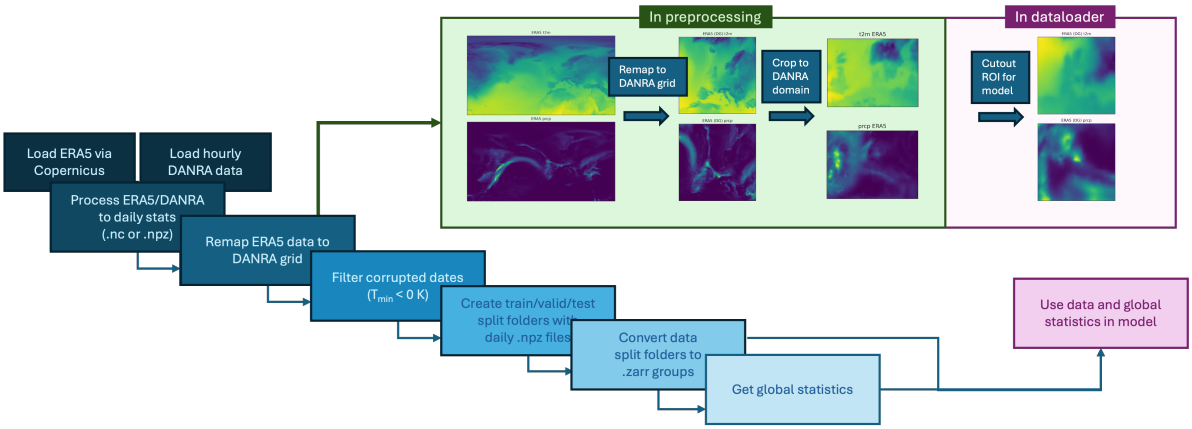
5. *Conditional LR channels*: The LR precipitation is included using (a) LR normalized using LR statistics (LR-in-LR space), $x'_{\text{LR-in-LR}} = \frac{log(x+\epsilon)-\mu_{\text{LR}}}{\sigma_{\text{LR}}}$, (b) LR normalized using HR statistics (LR-in-HR space), $x'_{\text{LR-in-HR}} = \frac{log(x+\epsilon)-\mu_{\text{HR}}}{\sigma_{\text{HR}}}$, and (c) using the upsampled LR field as the residual baseline (with LR-in-HR space as default baseline). This pairing exposes systematic LR-HR biases directly during training.

6. *Temporal splits*: We split the data in training ([1991-2015]), validation([2016-2018]), and test([2019-2020]) splits to maintain independence across all experiments, and to be able to evaluate the model's performance on unseen data and generalizability skill.

7. *Quality filtering*: A small number of corrupted days (e.g. negative precipitation artefacts) are removed, only for the final evaluation year, 2020. Masking of ocean and near-coast pixels is documented and applied consistently.

S2.3 Scaling and conditioning statistics

To ensure strict reproducibility, all global means and standard deviations used for log-z-score normalization are stored in metadata files and loaded during training and inference. The dual-LR conditioning mentioned above requires two separate sets of these statistics, ensuring correct mapping between ERA5 and DANRA distributions. As described, since precipitation exhibits a strongly skewed, heavy-tailed distribution with immense zero-inflation and an extended upper tail, training directly on this data in physical space leads to highly unbalanced gradients, where rare extreme events dominate the loss, and near-zero regimes collapse into numerical under-resolution. To mitigate this we apply the log-transformation described in Equation 17. This logarithmic transformation compresses the upper tail and spreads near-zero values, improving both numerical conditioning and stabilizing the optimization. Similar log-transformations are widely used across multiple fields like hydrological modelling, precipitation post-processing, and machine learning approaches to rainfall estimation, not unlike



Supplemental Information, Figure S1: *Data preprocessing and training pipeline.* Flowchart summarizing the data preparation steps: regridding ERA5 to DANRA grid, co-location, cutout generation, scaling and dataset splitting. Ensures reproducibility of the preprocessing workflow.

Table S1: Statistics overview for training and complete (all = train + validation + test) dataset. Statistics are used during data transformation and scaling for training stability (data closer to normally distributed), and reversely for transforming data back to physical space.

	mean μ [mm]	**st. dev.** σ [mm]	**min** [mm]	**max** [mm]	**log-mean** $\mu_{\log}$	**log-st. dev.** $\sigma_{\log}$	**min$_{\log}$**	**max$_{\log}$**
DANRA (train)	2.239	4.4272	1×10^{-10}	175.6386	-4.3011	7.2432	-23.0259	5.1684
DANRA (all)	2.2444	4.4291	1×10^{-10}	176.7636	-4.2858	7.2392	-23.0259	5.1748
ERA5 (train)	2.3718	3.9279	3.0304×10^{-11}	86.5126	-2.4745	5.8714	-24.2197	4.4603
ERA5 (all)	2.3961	3.9634	3.0304×10^{-10}	86.5126	-2.4946	5.9206	-24.2197	4.4603

our own work (Wu et al., 2019; Chu et al., 2022). Standardization by global mean and standard deviation further ensures approximate unit variance across the training set, which is consistent with the variance-stabilizing preconditioning philosophy of EDM (Karras et al., 2022). Importantly, we compute the normalization parameters, $\log(\mu)$ and $\log(\sigma)$, using training data only and store them as meta data to ensure strict reproducibility and to avoid information leakage across temporal splits. During inference and evaluation, all samples are back-transformed with the inverse transformation, using the same stored statistics. We do note that while other alternative transformations (Box-Cox, quantile normalization) could be considered and improve training and model performance, we see that the log-z-score transformation provides a simple compromise between variance stabilization and preservation of precipitation structure. Though, In future research, it would be of interest to see whether a more sophisticated transformation might be helpful in stabilising the model's performance on extremes and accumulation.

S3 Full Hyperparameter and Environment Specifications

This section documents the complete set of architectural, training, sampling, and environment parameters used in all experiments, complementing the high-level description in Sections 2 and 3.

All experiments were run using a configuration-driven pipeline, with YAML configuration files defining model, data, training, sampling, and evaluation behaviour, and Slurm + Singularity used for execution on the LUMI supercomputer. All experiments were developed and executed on the LUMI supercomputer (LUMI, 2022), using AMD GPU nodes. A typical full workflow - including training, sampling, and evaluation with a 32-member ensemble - runs on a single LUMI-G node equipped with eight MI250X GPUs, 56 CPU cores, and 60 GB of memory per GPU, within a six-hour time limit. Additional computational resources were required for extensive sampler and architecture component experiments as well as for running the model with the auxiliary RainGate module. These are not included in this configuration which is representative for a run of the final selected model.

Table S2: Conditioning inputs used across experiments. All conditioning fields are spatially aligned with the HR target before being passed to the network.

Condition type	Variables	Resolution	Used in B0	Notes
LR meteorology (ERA5)	Precipitation (main)	LR → aligned to HR	Yes	Additional ERA5 variables available in pipeline.
LR meteorology (optional)	CAPE, EWVF, NWVF, MSLP, temperature, geopotential heights at 250/500/850/1000 hPa	LR → aligned to HR	No	Included in some experiments depending on YAML config.
Geography	Land–sea mask, topography	HR grid	No	Enabled in geographic-conditioning architecture/final model (if retained).
Seasonality	Sinusoidal Day-Of-Year (DOY) encoding	Scalar / broadcast	No	Applied via FiLM modulation in decoder when enabled.
Dual-LR precipitation	Two precipitation channels normalized in LR-space and HR-space	LR → aligned to HR	No	Tested (dual-normalization); not retained due to degraded performance.

All experiment configurations are defined through version-controlled YAML files. Complete configuration files for baseline (B0), final model (F = B1_GSDF_RGBCE), and all architecture variants (B0_xxxx or B1_xxxx) are publicly available in the project repository and archived on Zenodo (see Code and Data Availability section). This section presents the most reproducibility relevant parameters and choices for the model, presented in tables.

Table S3: Main architectural configuration for the EDM-based model.

Component	Configuration/Settings
Core	**Architecture type**: UNet with encoder-decoder structure **Preconditioning**: EDM-style preconditioning wrapper (EDMPrecondUnet) **Output parameterisation**: direct prediction of $\hat{x}_0$ **Residual prediction**: Configurable (predict_residual), disabled in baseline B0 **Spatial dimensionality**: 2D fields **Input channels**: Variable, 3 (LR precip. + land-sea mask + topography) in baseline B0 **Output channels**: 1 (single target variable, precipitation in main experiments)
Encoder/ decoder	**Downsampling blocks**: 4 **Block depths**: [2, 2, 2, 2] **Final feature map channels**: 512 **Upsampling method**: transposed convolution (use_resize_conv = false) **Decoder normalisation**: Group Normalisation. **Number of groups**: 8. **Activation function**: SiLU
Attention	**Multi-head self-attention**: enabled **Block depths**: [2, 2, 2, 2] **Final feature map channels**: 512
Embedding	**EDM noise-level embedding** via learned Multi-Layer Perceptron (SigmaEmbed) **Time embedding dimensions**: 256 **Seasonal Conditioning** applied via FiLM-style modulation in the decoder

Table S4: Architectural and conditioning variants evaluated in component experiments relative to baseline B0.

Component	Setting	Notes / Implementation Details
Residual prediction (_R)	Predict $\Delta x = x_{HR} - x_{LR\uparrow}$	Replaces direct $\hat{x}_0$ target with residual target (Eq. S7–S8).
Dual-LR conditioning (_D)	LR-in-LR and LR-in-HR normalization	Two LR precipitation channels normalized with different statistics (Sec. S1.5).
Geographical conditioning (_G)	Land–sea mask + topography	Static HR fields concatenated to input channels. Enables spatial awareness of coastline and orography.
SDF-weighted loss (_SDF/_GSDF)	Spatial loss weighting $w(i, j)$	Distance-to-coastline weighting (Eq. S9–S11) emphasizing land and near-coastal pixels. Optionally normalized per sample.
EMA (_EMA)	Exponential Moving Average of weights	Shadow weights updated during training. Used only in EMA architecture experiments; not retained in final model due to seed-sensitive behaviour.
RainGate auxiliary head (_RGBCE)	Binary wet/dry classification	Auxiliary BCE loss for precipitation occurrence ($>$ 0.1 mm day^{-}1). Trained jointly; does not gate sampling.
Classifier-Free Guidance (CFG)	Conditional dropout guidance	Available in framework; empirically degraded results and therefore disabled in main experiments.

Table S5: Training hyperparameters for the EDM-based model.

Category	Parameter	Value	Notes
Optimisation	Optimiser	AdamW	
	Learning rate	2×10^{-4}	
	Weight decay	1×10^{-6}	
	Batch size	16	
	Epochs (max)	400	Early stopping enabled.
LR scheduling	Scheduler	ReduceLROnPlateau	
	Reduction factor	0.5	
	Patience	15 epochs	
	Minimum LR	1×10^{-6}	
Regularisation / stability	Gradient clipping	Disabled	
	Mixed precision	Disabled	
	EMA	Disabled (B0)	Enabled only in EMA experiments.
	Early stopping	Enabled	Patience 75, min $\Delta = 10^{-4}$.
Initialisation	Weight init	Xavier uniform	
	Random seed	504	Additional seeds used in architecture component experiments (see Tables S4–S5).

Table S6: EDM parameters (training and sampling).

Category	Parameter	Value	Notes
Training time	P_{mean}	-1.5	Log-normal noise level distribution (EDM).
	P_{std}	1.2	
	σ_{data}	1.0	Baseline B0. Tuned for final model runs along with sampler params.
	Loss target	$\hat{x}_0$	Direct prediction of the denoised sample.
Sampling time	Sampler type	EDM Heun	Karras noise schedule; second-order Heun updates.
	$\sigma_{\min}$	0.002	Minimum noise level.
	$\sigma_{\max}$	80	Maximum noise level.
	ρ and steps N	$\rho = 7.0$, $N = 56$	Schedule exponent and number of sampling steps.
Scale-aware control	Churn (S_{churn}, $S_{\min}$, $S_{\max}$, S_{noise})	2.0, 40, 80, 1.0	Optional stochasticity; set to zero for baseline/component experiment runs unless stated.
	σ^* default	1.0	Global scaling used for baseline/component experiments.
	Mode	Global scaling	Applies σ^* to the full trajectory.
	Optional late-step ramp	60–85%	Smoothstep interpolation (if enabled).

Table S7: Benchmark methods and parameters. Benchmark methods are bilinearly upsampled-to-HR resolution ERA5 fields (bilinear), and bilinearly upsampled and quantile mapped ERA5 fields (QM)

Benchmark	Trainable	Parameters	Notes
Bilinear ERA5	No	–	Deterministic interpolation baseline.
Quantile mapping (QM)	No	15 quantile pairs; wet-day threshold 0.5 mm day^{-}1	Enforces marginal distribution matching by construction.

Table S8: Generation and evaluation defaults used throughout experiments unless otherwise stated.

Stage	Parameter	Value	Notes
Generation	Sampler	EDM Heun	Karras schedule; churn/CFG optional depending on config.
	Generation batch size	16	Matches training batch size.
	Output domain	128 × 128 HR cutout	Stationary cutout around Denmark.
	Ensemble size M	32	Main experiments.
	Samples per input	Configurable	Typically full validation/test datasets.
	Classifier-free guidance (CFG)	Disabled	Available; empirically worsened results.
	EMA-based sampling	Disabled	Available; not used in main experiments.
Evaluation	Split	Validation or held-out test	Val used for component study/sampler tuning; test for final reporting.
	Units	Physical units	Metrics computed after inverse normalization using training-split statistics.

Table S9: Computational environment used for experiments.

Category	Component	Specification / Notes
Hardware	System	LUMI supercomputer.
Hardware	GPUs	AMD MI250X, 8 GPUs per node.
Hardware	CPUs	56 CPU cores per node.
Hardware	Memory	60 GB per GPU.
Software	Execution	Slurm + Singularity.
Software	Framework	PyTorch-based implementation.
Reproducibility	Seeding	Deterministic seeding for data loaders and sampling.
Reproducibility	Configs	YAML configurations version-controlled.

S4 Complete architectural component study

This section focuses on expanding on the summary of architectural component experiments presented in Tables S10 and S11, and in Figure Table S2. In the following we will detail the design, performance implications and interactions between (some) of the individual architecture components.

All architectural component experiments are conducted using a fixed EDM sampler configuration. The sampler parameters are chosen to yield stable and physically plausible samples across all tested architectures, but are not tuned for any specific configuration. This ensures that performance differences across architectures can be attributed to architectural choices rather than differences in sampling dynamics.

S4.1 Experimental design

Our full experimental workflow follows these steps:

1. → **Minimal, stable EDM**: We fix a stable EDM sampler configuration, which is used *unchanged* for all architectures.

2. → **Architectural and conditioning experiments**: We perform all architecture and conditioning studies under the selected fixed sampler from above.

3. → **Selected final architecture**: Based on the full component experiment suite, we select a final model architecture, F, that performs best under the desired evaluation metrics.

4. → **Tune sampler hyperparameters**: Only for the final model F, we tune the sampler hyperparameters $(\rho, S_{\text{churn}}, \sigma)$.

5. → **Evaluate F**: With the final selected optimal sampler hyperparameters (ρ^{optimal}, $S_{\text{churn}}^{\text{optimal}}$, σ^{optimal}) we run the full evaluation suite with the optimized model $\text{F}^{\text{optimal}}$ on the held-out test period.

6. → **Analyse** σ^*: We finally analyse σ^* as an inference-time control only on $\text{F}^{\text{optimal}}$.

S4.1.1 Reference sampler used for component experiments

All results from the architectural component experiments (Tables S10 and S11) are generated using the same fixed EDM sampler configuration. The parameters are listed below for completeness:

- **Sampling steps**: $N = 56$. **Noise distribution**: $P_{\text{mean}} = -1.5$, $P_{\text{std}} = 1.2$. **Noise schedule**: $\sigma_{\text{min}} = 0.002$, $\sigma_{\text{max}} = 80$, $\rho = 7.0$. **Stochastic churn**: $S_{\text{churn}} = 2.0$, $S_{\text{min}} = 40$, $S_{\text{max}} = 80$, $S_{\text{noise}} = 1.0$. **Scale-aware analysis**: $\sigma^* = 1.0$ (no noise-scale perturbation during component experiments)

Residual prediction and dual-normalization baselines are disabled unless explicitly stated as part of an experiment.

S4.2 Sampler hyperparameter sweep

With the sampler-hyperparameter sweep, we can investigate the influence of stochasticity on our model's performance, see Figure Table S2. These results show us that there is a clear monotonic regime structure in sampler hyperparameters: our sweep does not produce chaotic behaviour, it produces structured, interpretable gradients in performance. We see that increasing stochasticity through higher S_{churn} and/or σ scaling, increases extremes, yearly asun and wet-day frequency, while slightly flattening the PSD slope, which reflects more small-scale variability. On the other hand, lowering the stochastic parameters leads to more conservative fields with reduced extremes and steeper PSD slopes. This illustrates that the sampler acts as a controllable bias-variance regulator, and points to the importance of tuning these hyperparameters both pre-training, and during validation. Our specific selected sampler (red box) lies in a balanced, somewhat-interior domain. We have prioritized good capture of extremes and good stochastic performance, leading us to select a model where PSD is close to DANRA, CRPS is high-performing, wet-day frequency is closed to observed, and yearly sum is increased compared to low-stochastic runs. A few models cover these criteria, but we have selected the sampler configuration that ends up with extremes closest to the target DANRA distribution. Relative to architectural changes, sampler hyperparameters induce more monotonic and easily interpretable trade-offs, making them a useful control manifold for understanding bias-variance behaviour. We also see that trade-offs based on sampler configuration are metric-specific. Generally, bulk metrics, CRPS, PSD, ISS, are robust, whereas tail and accumulation metrics are more sensitive - which we also observe in the architectural component experiments. The diffusion sampling trajectory is behaving consistently with theoretical expectations: changing injected noise changes effective correlation length, high-frequency power, and tail-heaviness. The sampler hyperparameters selected define a smooth control manifold for investigating the effects of σ^*.

S4.3 σ^*: scale-aware inference control

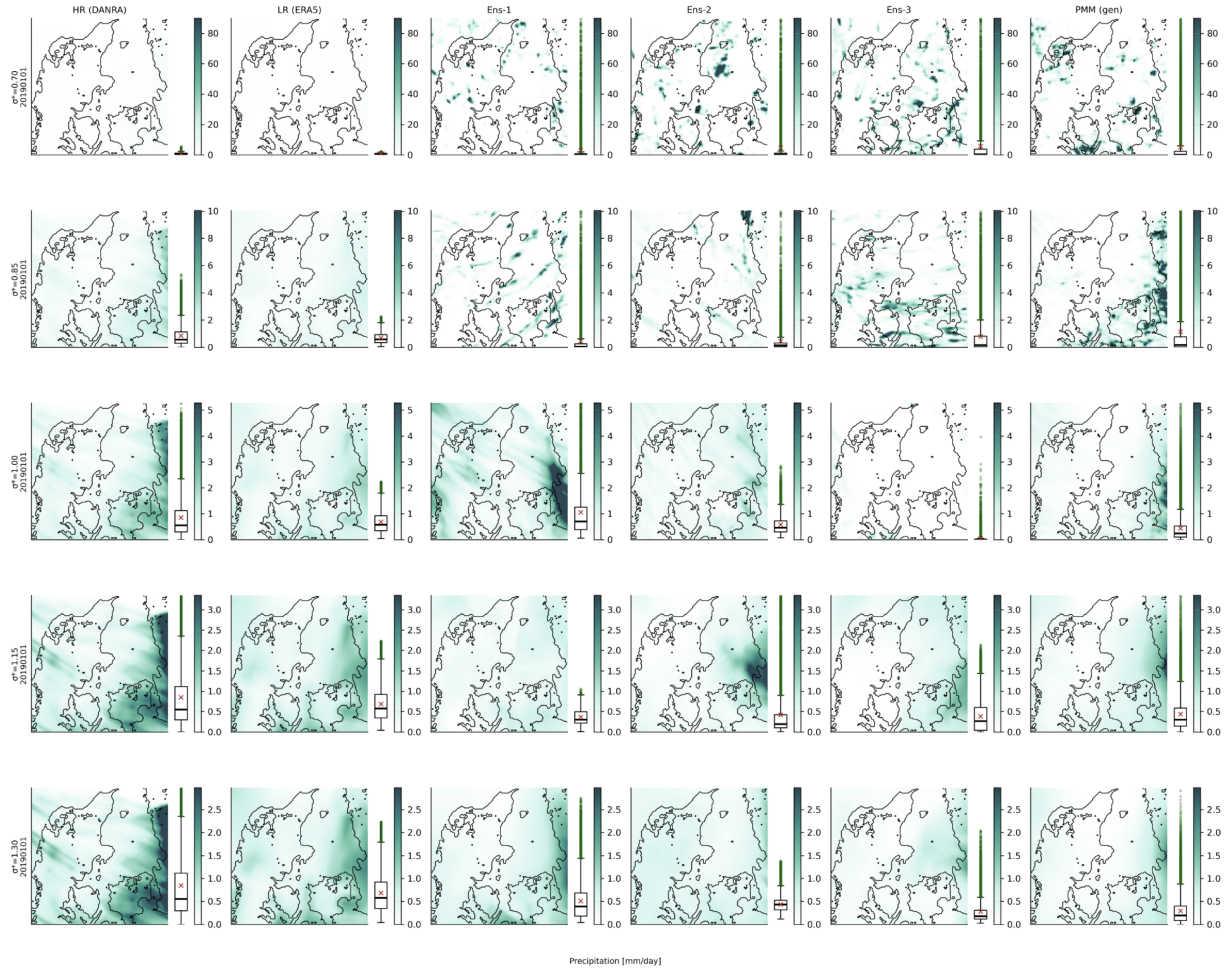
All σ^* analyses are performed on the final selected architecture with its tuned sampler configuration. σ^* is not used during architectural experiments and is not part of the sampler hyperparameter optimization. A broader exploratory sweep across σ^* values ($\sigma^* = [0.70, 0.85, 1.00, 1.15, 1.30]$) was tested first to identify a stable and physically meaningful range. This first analysis allowed us to narrow the viable regime for the main analysis, ending with a region of $\sigma^* \approx 0.90 - 1.15$, which is presented in Main Section 4.5.

The σ^* sweep is not intended to define an operational control parameter, but to investigate and present the stability regime of EDM-based downscaling. Values of σ^* far from 1 correspond to sampling trajectories not encountered during training, and therefore represent, controlled, but out-of-distribution behaviour. The results from the first sweep show that within a restricted range (approximately 0.9-1.1 in this study), σ^* produces monotonic and interpretable changes in spatial variability and ensemble spread. Outside of this range, spectral diagnostics reveal excess small-scale noise and collapsed variability, which clearly indicates loss of physicality.

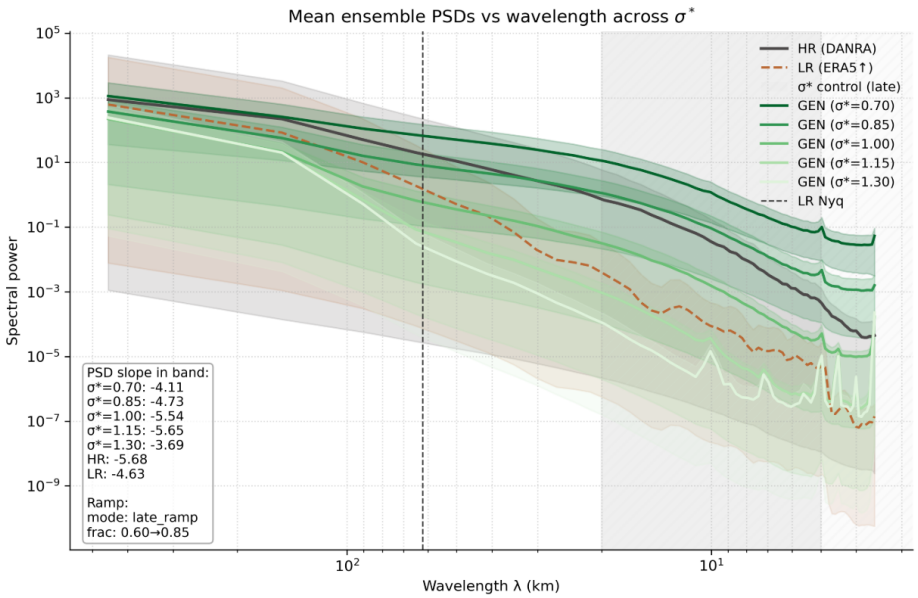
Within the smaller stable regime (approximately $\sigma^* \approx 0.9 - 1.1$), all diagnostics vary smoothly and monotonically with σ^*. Scale-aware correlation increases as σ^* decreases, re-

Rho	S_churn	Sigma scal	PSD slope (high-k)	PSD interce pt (high-k)	ISS at 20km >= 10 mm/day	CRPS (mean)	P99	P99.9	Wet-day Frequency	Yearly sum (2017)	STD on yearly sum (2017)
		DANRA	-2,29	-3,12	1,00	0,00	18,41	34,34	0,33	834,13	116,12
5.0	0.0	0,9	-2,406	-3,429	0,7831	0,775	14,13	30,14	0,2956	623,34	71,43
5.0	0.0	0,95	-2,398	-3,409	0,7843	0,768	14,34	30,51	0,2985	634,56	72,14
5.0	0.0	1	-2,392	-3,391	0,7845	0,761	14,53	30,86	0,3010	644,82	72,79
5.0	0.0	1,05	-2,386	-3,376	0,7850	0,755	14,70	31,19	0,3033	654,23	73,40
5.0	0.0	1,1	-2,382	-3,361	0,7850	0,750	14,86	31,49	0,3054	662,91	73,97
5.0	0.5	0,9	-2,395	-3,404	0,7881	0,765	14,70	30,85	0,2986	643,03	76,51
5.0	0.5	0,95	-2,372	-3,391	0,7880	0,757	14,75	30,60	0,3022	650,95	78,74
5.0	0.5	1	-2,417	-3,389	**0,7905**	0,750	14,88	31,57	0,3048	662,28	75,06
5.0	0.5	1,05	-2,361	-3,360	0,7882	0,746	15,10	31,24	0,3067	669,66	80,05
5.0	0.5	1,1	-2,357	-3,347	0,7880	0,742	15,25	31,53	0,3086	677,90	80,64
5.0	1.0	0,9	-2,395	-3,389	**0,7902**	0,757	14,92	31,33	0,3016	655,52	77,36
5.0	1.0	0,95	-2,371	-3,374	0,7886	0,749	14,99	31,10	0,3052	664,49	80,03
5.0	1.0	1	-2,410	-3,369	0,7897	0,742	15,14	32,03	0,3080	675,83	76,35
5.0	1.0	1,05	-2,361	-3,345	0,7888	0,740	15,31	31,70	0,3094	682,25	81,31
5.0	1.0	1,1	-2,357	-3,333	0,7887	0,736	15,46	31,97	0,3111	690,06	81,89
5.0	2.0	0,9	-2,388	-3,355	0,7874	0,745	15,34	32,16	0,3071	678,49	79,13
5.0	2.0	0,95	-2,366	-3,341	0,7885	0,737	15,43	32,00	0,3107	689,33	**82,20**
5.0	2.0	1	-2,386	-3,324	0,7882	**0,730**	15,63	**32,83**	**0,3139**	700,69	78,55
5.0	2.0	1,05	-2,358	-3,315	0,7884	**0,729**	15,73	32,54	**0,3142**	705,32	**83,39**
5.0	2.0	1,1	-2,354	**-3,304**	0,7886	**0,726**	**15,86**	**32,79**	**0,3158**	**712,36**	**83,95**
7.0	0.0	0,9	-2,405	-3,428	0,7834	0,775	14,13	30,14	0,2956	623,39	71,41
7.0	0.0	0,95	-2,397	-3,408	0,7842	0,768	14,34	30,52	0,2985	634,61	72,13
7.0	0.0	1	-2,391	-3,391	0,7843	0,761	14,53	30,87	0,3010	644,87	72,79
7.0	0.0	1,05	-2,386	-3,375	0,7848	0,755	14,70	31,19	0,3033	654,29	73,40
7.0	0.0	1,1	-2,381	-3,360	0,7846	0,750	14,86	31,48	0,3054	662,95	73,96
7.0	0.5	0,9	-2,371	-3,389	0,7856	0,765	14,60	30,90	0,2981	643,52	76,79
7.0	0.5	0,95	-2,388	-3,385	0,7879	0,758	14,90	31,21	0,3013	653,89	77,29
7.0	0.5	1	-2,382	-3,369	0,7880	0,753	15,09	31,54	0,3037	663,77	78,01
7.0	0.5	1,05	-2,377	-3,354	0,7881	0,748	15,25	31,83	0,3058	672,81	78,68
7.0	0.5	1,1	-2,372	-3,340	0,7884	0,743	15,41	32,12	0,3077	681,14	79,29
7.0	1.0	0,9	-2,362	-3,370	0,7843	0,758	14,79	31,14	0,3007	653,93	77,59
7.0	1.0	0,95	-2,388	-3,371	**0,7903**	0,751	15,11	31,68	0,3041	665,91	78,12
7.0	1.0	1	-2,383	-3,355	0,7901	0,746	15,29	32,01	0,3063	675,37	78,81
7.0	1.0	1,05	-2,378	-3,341	**0,7903**	0,742	15,45	32,30	0,3083	684,00	79,46
7.0	1.0	1,1	-2,373	-3,329	**0,7904**	0,738	15,60	32,57	0,3101	691,95	80,05
7.0	2.0	0,9	-2,353	-3,340	0,7822	0,746	15,15	31,64	0,3055	673,51	78,99
7.0	2.0	0,95	-2,382	-3,339	0,7870	0,740	15,51	32,48	0,3093	688,01	79,83
7.0	2.0	1	-2,377	-3,326	0,7871	0,736	15,68	32,78	0,3112	696,65	80,47
7.0	2.0	1,05	-2,373	-3,313	0,7868	0,732	15,82	**33,04**	0,3130	**704,52**	81,06
7.0	2.0	1,1	-2,369	**-3,302**	0,7870	**0,729**	**15,96**	**33,29**	**0,3146**	**711,77**	**81,60**
9.0	0.0	0,9	-2,404	-3,427	0,7833	0,775	14,13	30,14	0,2957	623,47	71,42
9.0	0.0	0,95	-2,397	-3,408	0,7842	0,768	14,34	30,52	0,2985	634,69	72,13
9.0	0.0	1	-2,391	-3,390	0,7843	0,761	14,53	30,88	0,3011	644,95	72,79
9.0	0.0	1,05	-2,385	-3,375	0,7849	0,755	14,70	31,19	0,3034	654,36	73,40
9.0	0.0	1,1	-2,380	-3,360	0,7849	0,750	14,86	31,49	0,3054	663,03	73,96
9.0	0.5	0,9	-2,371	-3,389	0,7861	0,765	14,60	30,91	0,2981	643,60	76,79
9.0	0.5	0,95	-2,365	-3,370	0,7863	0,758	14,81	31,28	0,3008	654,55	77,60
9.0	0.5	1	-2,359	-3,354	0,7866	0,752	14,99	31,62	0,3031	664,55	78,34
9.0	0.5	1,05	-2,354	-3,339	0,7864	0,747	15,17	31,93	0,3053	673,70	79,02
9.0	0.5	1,1	-2,349	-3,325	0,7862	0,743	15,32	32,21	0,3072	682,13	79,65
9.0	1.0	0,9	-2,362	-3,370	0,7839	0,758	14,79	31,14	0,3007	654,00	77,59
9.0	1.0	0,95	-2,356	-3,352	0,7844	0,752	14,99	31,50	0,3032	664,55	78,37
9.0	1.0	1	-2,351	-3,337	0,7842	0,746	15,17	31,83	0,3055	674,18	79,09
9.0	1.0	1,05	-2,346	-3,322	0,7846	0,742	15,33	32,12	0,3075	682,96	79,74
9.0	1.0	1,1	-2,341	-3,309	0,7848	0,738	15,48	32,40	0,3094	691,08	80,34
9.0	2.0	0,9	-2,352	-3,340	0,7822	0,746	15,16	31,65	0,3055	673,59	79,00
9.0	2.0	0,95	-2,347	-3,324	0,7828	0,741	15,34	31,99	0,3078	683,37	79,73
9.0	2.0	1	-2,343	-3,310	0,7830	0,737	15,51	32,30	0,3098	692,28	80,39
9.0	2.0	1,05	**-2,338**	**-3,297**	0,7828	0,733	15,66	**32,56**	0,3116	700,39	81,02
9.0	2.0	1,1	**-2,334**	**-3,285**	0,7832	**0,730**	**15,80**	**32,82**	0,3132	**707,89**	**81,58**

Supplemental Information, Figure S2: *Results from the full sampler hyperparameter sweep.* Selected sampler for final evaluation on test data split is framed with a red box.



Supplemental Information, Figure S3: *Visual effect of different σ^*, edge cases, validation dataset.*



Supplemental Information, Figure S4: *Scale-aware inference edge full PSD spectrum, edge cases, validation dataset.*

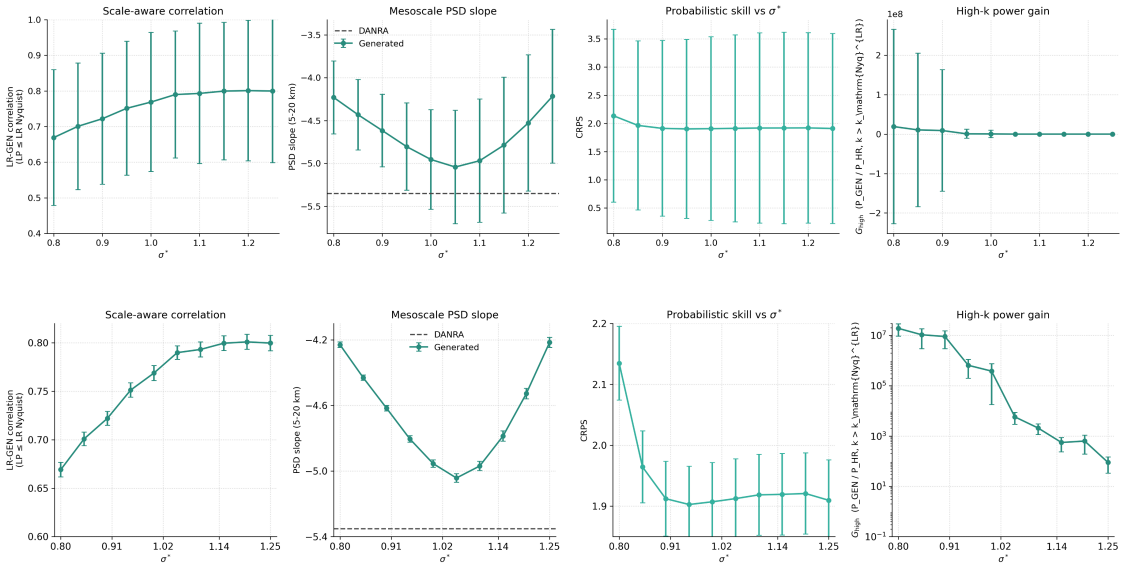
flecting increased small-scale variability. Conversely, larger σ^* values suppress high-frequency structure and increase spatial smoothness, consistent with the theoretical balance between stochastic forcing and denoising drift (Section S1.7). Figure S5 shows the same mean values, but with different error represented. The top row illustrates the standard deviation on the full two-year test dataset, with 32 ensemble members per date, exhibiting a large spread due to inter-ensemble and intra-annual variability. To better illustrate the actual mean behaviour of the metrics, given a changing σ^*, the bottom row shows the mean values of the four metrics, with the Standard Error on the Mean (SEM) instead of the full standard deviation.

S4.4 Interpretation of architecture and conditioning choices

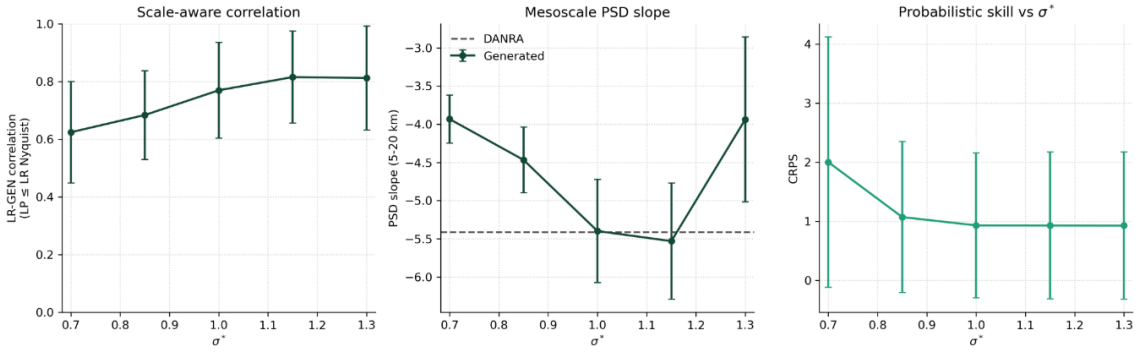
Tables S10 and S11 show the full investigation of model performance given architectural and conditioning changes, as well as seed variability.

Across the B0 architectural component experiments, EMA does not produce a consistent reduction in seed-to-seed spread; instead it shifts model behaviour in a configuration- and seed-dependent way, sometimes improving specific metrics while degrading others. Across the results, we see that a dominating factor for model behaviour is the conditioning strategy rather than secondary architectural modifications - though the combination of both conditioning strategy and architecture can yield the most stable improvements across seeds.

For several decision-relevant diagnostics, the within-configuration spread across seeds is of the same order of differences between neighbouring architectural variants, which makes single-seed comparisons unreliable. These findings clearly motivate selecting configurations on consistent direction improvements across seeds and reporting mean performance together



Supplemental Information, Figure S5: *Standard deviation (top) and standard error of the mean (bottom) on scale-aware metrics.*



Supplemental Information, Figure S6: *Scale-aware inference analysis: edge cases, validation dataset.*

with seed spread. Extremes like P99.9 and yearly accumulation are disproportionately sensitive because propagated small daily biases and upper-quantile shifts compound over long aggregation and rare-event statistics. This discloses the stochastic sensitivity of the model, and again highlights the difficulty on correctly representing the underdetermined extreme tail behaviour - also with sophisticated generative models.

This pattern of stable bulk metrics but variable tails is generally consistent with a moderately flat loss landscape in which multiple near-equivalent local minima yield similar bulk performance but differ in how the upper tail and accumulation behaviour are calibrated. These findings motivate selecting configurations based on consistent directional improvements *across seeds* and reporting mean performance together with seed spread, rather than relying on single-seed outcomes, especially when decisions depend on extremes and accumulated quantiles.

Table S10: *Overview of model performance given various architectural, conditioning choices, and training seed.*

	PSD slope ($\lambda < 20$ km)	**ISS at 20 km** (≥ 10 $\frac{\text{mm}}{\text{day}}$)	**CRPS** (mean)	**P99** [$\frac{\text{mm}}{\text{day}}$]	**P99.9** [$\frac{\text{mm}}{\text{day}}$]	**Wet-day frequency** [%]	**Yearly sum** (2017) [mm]	**STD on yearly sum** [mm]
DANRA	**-2.29**	**-**	**-**	**18.41**	**34.34**	**33.28**	**834.13**	**116.12**
B0 (504)	-2.46	0.7893	0.759	15.67	31.79	31.25	699.03	55.81
B0_1 (1011)	-2.73	0.7848	0.864	15.28	36.66	25.81	582.53	38.85
B0_2 (607)	-2.56	0.7807	0.787	15.87	34.37	30.19	677.30	50.09
B0_EMA (504)	-2.48	0.7876	0.758	16.20	33.70	30.01	699.35	53.93
B0_EMA_1 (1011)	-2.51	0.7931	0.774	16.41	36.40	30.30	693.17	54.21
B0_EMA_2 (607)	-2.47	0.7924	0.757	16.26	34.34	30.97	712.80	56.03
B0_S (504)	-2.36	0.7815	0.740	15.37	31.16	31.87	698.58	55.16
B0_S_1 (1011)	-2.36	0.7905	0.731	16.03	33.71	30.64	702.33	53.75
B0_S_EMA (504)	-2.33	0.7905	0.699	15.72	31.20	33.41	739.73	53.85
B0_S_EMA_1 (1011)	-2.31	0.7925	0.724	18.31	36.25	33.91	817.80	63.85
B0_G (504)	-2.45	0.7915	0.767	16.16	34.97	30.95	695.67	75.83
B0_G_1 (1011)	-2.45	0.7932	0.749	16.65	34.21	30.76	720.11	73.87
B0_GSDF (504)	-2.45	0.7853	0.753	16.32	33.48	31.10	722.08	74.98
B0_GSDF_1 (1011)	-2.50	0.7954	0.756	15.96	33.45	31.43	707.26	81.45
B0_RGBCE (504)	-2.37	0.7888	0.760	15.65	31.28	31.84	704.55	53.30
B0_RGBCE_1 (1011)	-2.62	0.7913	0.770	17.20	39.10	29.70	710.84	51.02
B0_D (504)	-2.34	0.7883	0.795	13.71	27.45	30.97	636.30	52.69
B0_LRinHR (504)	-2.42	0.7881	0.757	15.97	33.34	30.78	699.62	61.68
B0_R_HR (504)	-2.36	0.7372	1.088	9.42	20.96	20.93	371.03	33.83
B0_R_LR (504)	-2.42	0.7880	0.7629	15.85	33.22	30.25	684.29	60.94

Table S11: *Overview of model performance given various architectural, conditioning choices, and training seed.*

	PSD slope ($\lambda < 20$ km)	**ISS at 20 km** (≥ 10 $\frac{\text{mm}}{\text{day}}$)	**CRPS** (mean)	**P99** [$\frac{\text{mm}}{\text{day}}$]	**P99.9** [$\frac{\text{mm}}{\text{day}}$]	**Wet-day frequency** [%]	**Yearly sum** (2017) [mm]	**STD on yearly sum** [mm]
DANRA	**-2.29**	**-**	**-**	**18.41**	**18.41**	**33.28**	**834.13**	**116.12**
B1_G (504)	-2.34	0.7832	0.738	15.16	29.98	30.77	667.95	78.86
B1_G_1 (1011)	-2.42	0.7806	0.785	13.68	27.73	28.80	596.38	68.44
B1_GSDF (504)	-2.26	0.7888	0.719	15.56	30.47	33.00	725.12	75.55
B1_GSDF_1 (1011)	-2.35	0.7905	0.756	14.35	28.97	32.48	669.71	72.61
B1_RGBCE (504)	-2.33	0.7923	0.731	15.48	30.42	32.01	700.45	59.19
B1_RGBCE_1 (1011)	-2.45	0.7916	0.751	15.00	31.68	31.19	672.37	62.27
B1_G _RGBCE (504)	-2.26	0.7920	0.695	16.72	31.81	33.70	786.79	92.89
B1_G _RGBCE_1 (1011)	-2.39	0.7954	0.740	15.21	30.29	31.03	675.37	69.59
B1_GSDF _RGBCE (504)	-2.38	0.7871	0.736	15.68	32.78	31.12	696.65	80.47
B1_GSDF _RGBCE_1 (1011)	-2.38	0.7876	0.729	15.63	31.53	32.21	711.78	75.17

S5 Extended Evaluation Metrics

This appendix presents secondary diagnostics that complement the main text's ensemble, spatial-structure, and distributional metrics.

S5.1 Evaluation framework

We evaluate across a multi-pillar framework to allow for assessing model performance from multiple complementary angles. A more in-depth description of the different evaluation types is presented in the following:

1. **Probabilistic realism and ensemble calibration**
   (a) **Continuous Ranked Probability Score (CRPS)** (Hersbach, 2000) quantifies the integrated difference between the cumulative distribution functions of the ensemble forecast and the observed reference (DANRA). A perfectly reliable ensemble yields minimal CRPS and reproduces the observed distribution across quantiles.
   (b) CRPS is computed per date for the ensemble distribution and the Mean Absolute Error (MAE) for the individual ensemble members, enabling separation of ensemble spread from individual bias.
   (c) Additional diagnostics: **Probability Integral Transform (PIT)** and **rank histograms** (Hamill, 2001) are used to assess ensemble calibration and detect over-/under-dispersion.
2. **Spatial and scale-dependent structure** Accurate reconstruction of spatial organization is crucial for physical realism and hydrological applicability. We therefore evaluate:
   (a) **Intensity–Scale Skill (ISS)** (Casati et al., 2004) measuring how well intensity and pattern correlation are retained as precipitation fields are coarsened from fine to large scales. ISS is a better measure than the otherwise commonly used Fraction Skill Score (FSS) in very coastal areas when evaluating on land-only. ISS is evaluated at spatial neighbourhoods of [5, 10, 20, 40] km and thresholds of [1, 5, 10, 20, 50] $\mathrm{mm\,day^{-1}}$. The same metric is applied to the LR and QM benchmarks for reference.
   (b) **Power Spectral Density (PSD)** analysis (Lovejoy and Schertzer, 2013; Hess et al., 2025), which quantifies the spatial variance distribution as a function of wavelength. The slope of the isotropic PSD in log–log space provides a concise indicator of spatial scaling behavior and energy distribution. We exclude the lowest two k-bins to avoid domain-size effects.
3. **Statistical and climatological fidelity** We assess whether generated fields reproduce the observed precipitation statistics at different aggregation levels:
   (a) **Pixel-value distributions:** pooled over the entire test dataset to assess overall intensity representation and variance.

(b) **Annual accumulation:** Total annual precipitation accumulation

(c) **Tail and wet-day statistics:** upper quantiles (P95, P99, P99.9, P99.99) and wet-day frequency, to evaluate extreme-event realism.

Implementation-wise, pixel distributions are computed over all test days with seasonal summaries for 2019–2020.

4. **Controllable scale-awareness via σ^***

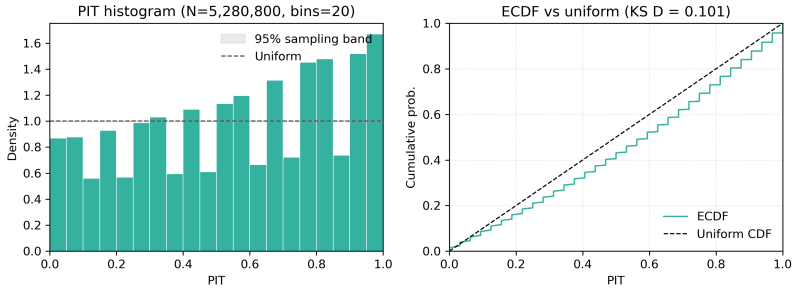
   (a) The controllability of the generative process is evaluated by varying σ^* during inference, analysing how spatial variability and ensemble spread respond.

   (b) For each σ^* value, we compute ensemble CRPS, correlation between Low-Resolution condition and Generated Sample, and PSD slope.

   (c) We evaluate PSD slopes over a mesoscale band (5–20 km), which is where we clearly observe the lack of power in ERA5. Matching the PSD slope in this band provides a physically grounded anchor for selecting σ^*.

S5.2 Rank and PIT histograms

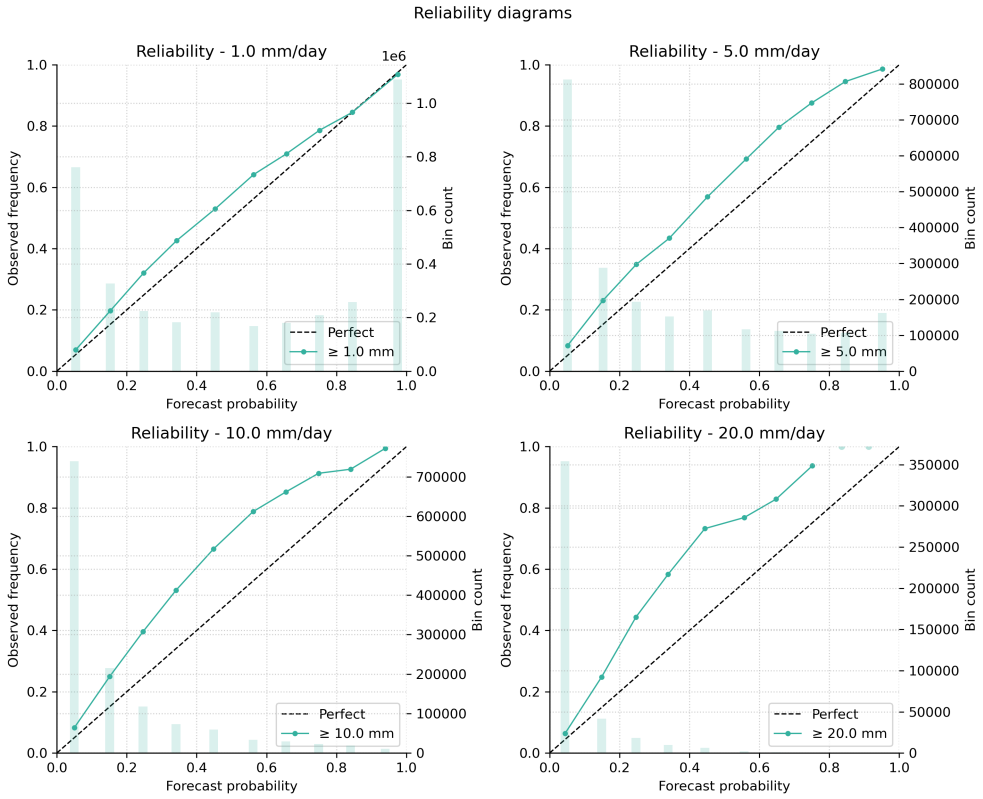
Figure S7 show PIT histogram and Empirical Cumulative Distribution Function (ECDF) for test years 2019-2020, from the final model run. The ensemble is broadly calibrated, but leans toward a dry bias (observations generally fall in the upper PIT bins) and exhibits slight underdispersion, consistent with what we see from the CRPS analysis.



Supplemental Information, Figure S7: *Ensemble calibration diagnostics: PIT histogram and ECDF.* Probability Integral Transform (PIT) histogram showing the EDM ensemble distribution, and its tendency to be too dry (HR sample most often located in the higher bins of PIT) and slightly underdispersed (slight U-curve).

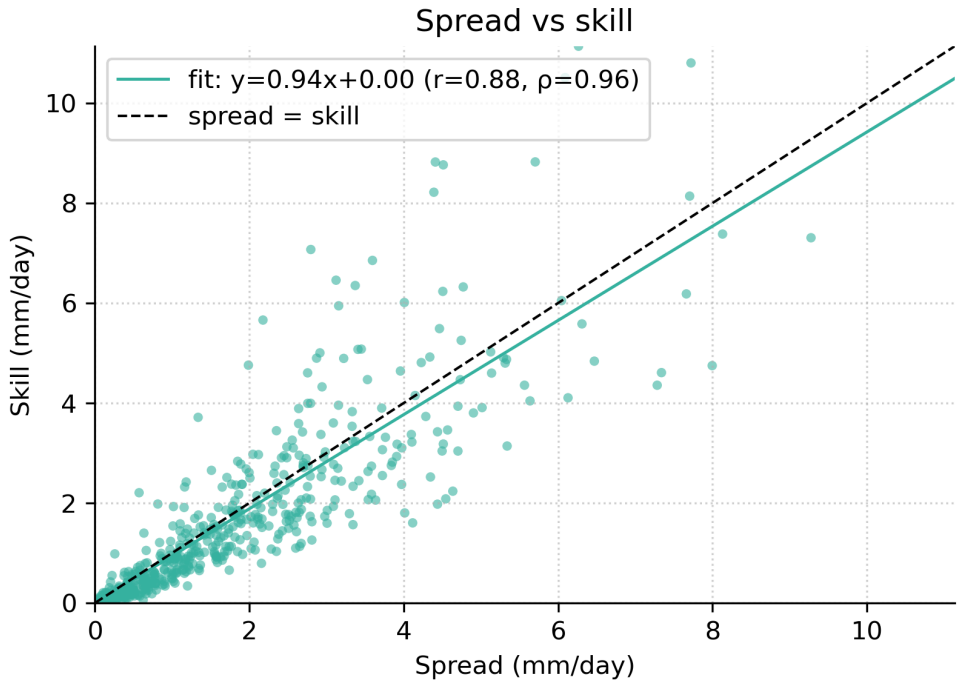
S5.3 Reliability and spread-skill

Reliability diagrams (Figure S8) show good calibration for light-to-moderate precipitation thresholds (1-5 mm day^{-}1) with deviations increasing for rarer events (10-20 mm day^{-}1), where sampling uncertainty and tail variability dominate. This behaviour is consistent with the greater sensitivity of extreme precipitation to stochastic training and sampling effects.



Supplemental Information, Figure S8: *Reliability diagrams for different rain event thresholds.*

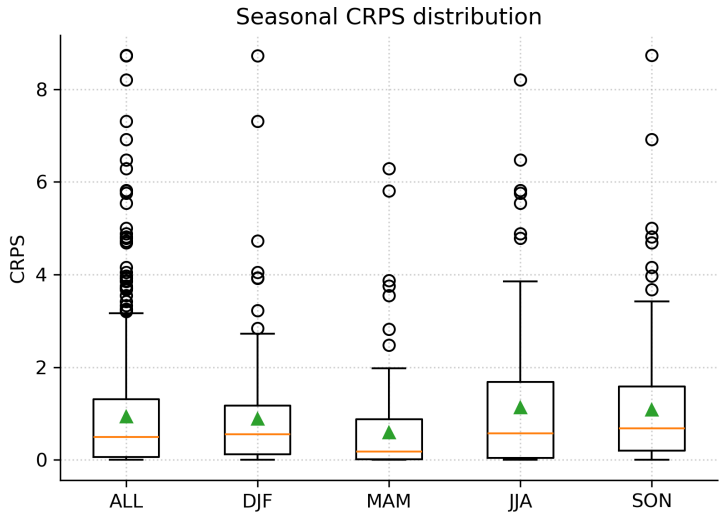
The spread-skill relationship in Figure S9 exhibits stong linearity (r = 0.88) and a fitted slope close to one (0.94). This indicates that the ensemble spread provides a meaningful proxy for the 'forecast' error. The slight slope deficit suggests mild under-dispersion at high spread values, which we have also seen in the PIT histogram, but oberall the ensemble is well calibrated in terms of uncertainty magnitude.



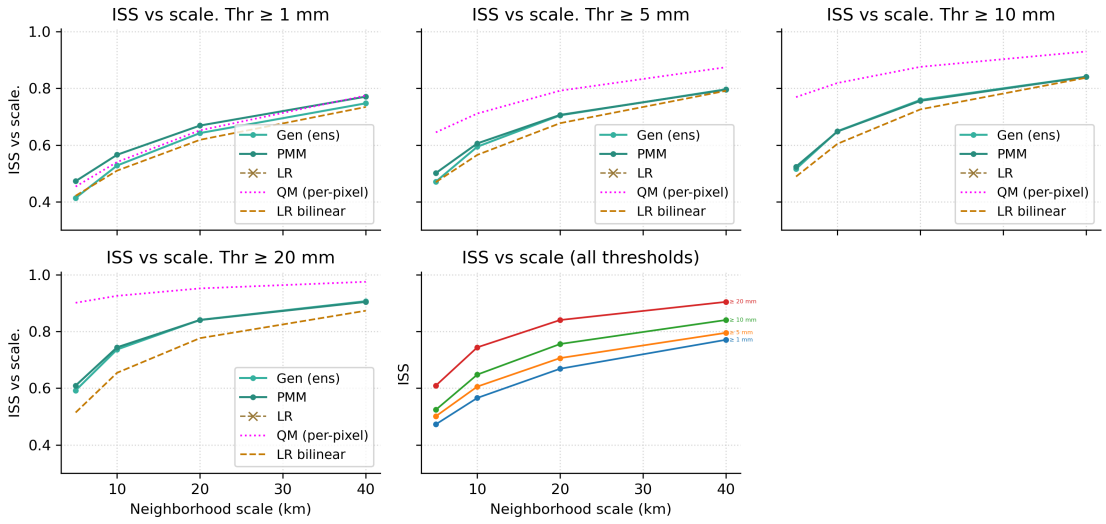
Supplemental Information, Figure S9: *Spread vs. skill.*

S5.4 Seasonal CRPS distribution

Figure S10 shows seasonal distributions as boxplots of daily CRPS values. For each day in the test period, CRPS is computed over the evaluation domain and aggregated into seasonal groups (DJF, MAM, JJA, SON). The boxplots display the interquartile range, median (orange line), mean (green triangle), and outliers. We see from the boxplots in Figure S10 that CRPS exhibits a some seasonal dependency, where highest mean and especially upper tail values occur during summer (JJA). This reflects the known difficulty of predicting or downscaling during convective regimes, which are strongly characterized by localized and intermittent precipitation. Winter (DJF) and spring (MAM) show comparatively lower spread, consistent with fewer extreme and convective-governed events. Across all seasons, the right-skewed distributions indicate that a relatively small number of high-impact days dominate the overall probabilistic error statistics.



Supplemental Information, Figure S10: *Seasonal distributions of the Continuous Ranked Probability Score (CRPS) for ensemble evaluation.* The orange lines correspond to the medians of each distribution, and the green triangle to the mean. Lower CRPS values indicate higher probabilistic accuracy.

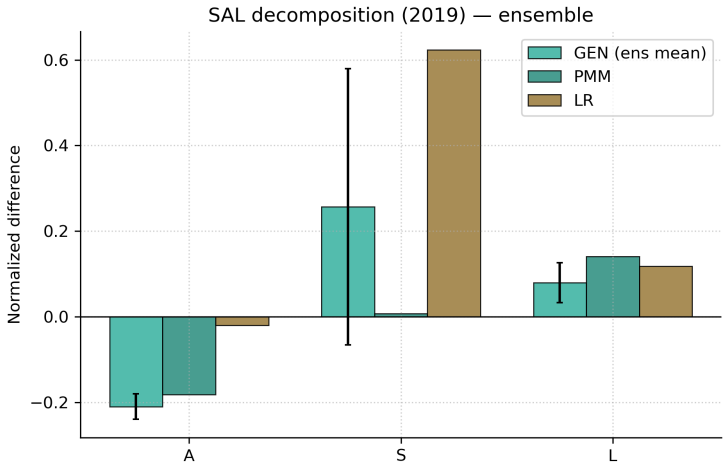


Supplemental Information, Figure S11: *Intensity Scale Skill across spatial scales and precipitation thresholds.* ISS as a function of threshold and spatial aggregation scale for the EDM-generated ensemble mean. High ISS values across scales indicate that the diffusion model preserves both the intensity distribution and spatial organization of precipitation.

S5.5 Spatial structure

dFigure S11 present the Intensity-Scale Skill Score (ISS) as a function of spatial aggregation scale and precipitation threshold. The quantile-mapped (QM) baseline consistently achieve high ISS values across scale and thresholds, often exceeding those of the EDM-generated ensemble. However, this behaviour should not be interpreted as superior spatial performance. ISS implicitly rewards marginal distribution matching and spatial smoothing, which are both enforced by construction in QM. At the same time, ISS is insensitive to event morphology, spatial coherence, and cross-scale dependence. As a result, ISS values for QM and low-resolution baselines are not directly comparable to those of generative models - that explicitly reconstruct spatial structure.

Figure S12 presents the Structure-Amplitude-Location (SAL) decomposition for the test year 2019. The low-resolution baseline exhibits a pronounced structural bias (high positive S), reflecting over-smoothed precipitation objects and lack of fine-scale variability. The diffusion-generated ensemble reduces that error, but with large variance from ensemble-member to ensemble-member. The PMM of the ensemble nearly eliminates all structure bias for this year. Amplitude differences are almost not exhisting for the LR benchmark, but the generated results exhibit a negative amplitude bias (negative A) which is explained by the general dry-bias that the model shows. Location errors are moderate across all products, with the generated ensemble members generally placing events better than the LR benchmark. Overall, the SAL decomposition confirms that the primary improvement by generative diffusion models lies in recovering realistic spatial structure rather than merely correcting domain-mean intensity.



Supplemental Information, Figure S12: *Structure-Amplitude-Location plots for 2019 test year.*

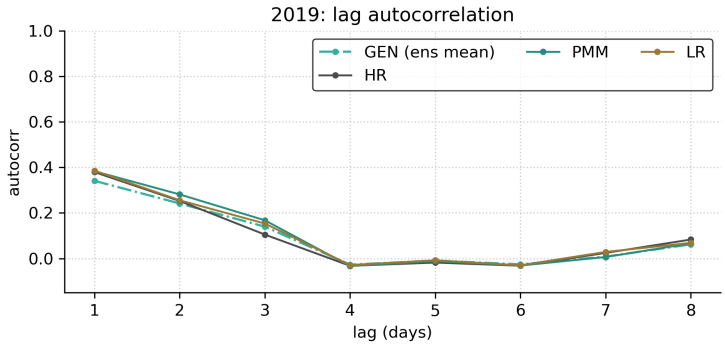
S5.6 Temporal statistics

The k-lag autocorrelation functions in Figure S13 demonstrate that the generated ensemble preserves the temporal memory structure of daily precipitation almost exactly. Particularly,

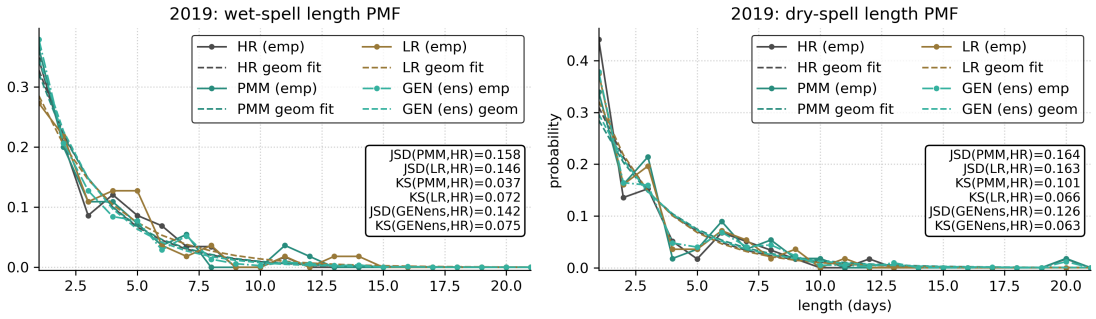
lag-1 and lag-2 correlations align closely with both HR and LR fields. This is most likely due to the implicit temporal dynamics learned through the LR ERA5 which works as a driver for the DANRA reanalysis, and as a condition for our generative model.

Dry-spell distributions are nearly indistinguishable across HR, LR and generated fields, indicating large-scale persistence properties are already well captured by the driving ERA5 fields and preserved by our downscaling model. Wet-spell lengths exhibit slightly larger deviations, with LR showing modest bias relative to HR, while the generated ensemble aligns slightly better with the HR reference. This points to our model preserving temporal characteristics without introducing artificial persistence or fragmentation - but without giving more insight into wet- and dry-spells than what the LR driver already gives.

It is important to note that the temporal analyses are computed over the full spatial domain and aggregated across all pixels. While this provides a robust assessment of large-scale temporal persistence, it does not capture local dry- or wet-spell characteristics. Local characteristics at e.g. catchment scale may exhibit stronger variability and greater hydrological relevance. Localized dry-spell persistence can, for example, critically influence soil moisture depletion and runoff generation, and such local effects should be analysed more in-depth before concluding on model performance across such impact-relevant metrics.



Supplemental Information, Figure S13: *k-lagged autocorrelations for test year 2019, EDM generated, HR DANRA and LR ERA5.*



Supplemental Information, Figure S14: *Wet- and dry-spell lengths for test year 2019, EDM generated, HR DANRA and LR ERA5.*

References

Brock, A., Donahue, J., and Simonyan, K.: Large Scale GAN Training for High Fidelity Natural Image Synthesis, https://doi.org/10.48550/ARXIV.1809.11096, 2018.

Casati, B., Ross, G., and Stephenson, D. B.: A New Intensity-Scale Approach for the Verification of Spatial Precipitation Forecasts, Meteorological Applications, 11, 141–154, https://doi.org/10.1017/S1350482704001239, 2004.

Chu, K.-S., Oh, C.-H., Choi, J.-R., and Kim, B.-S.: Estimation of Threshold Rainfall in Ungauged Areas Using Machine Learning, Water, 14, 859, https://doi.org/10.3390/w14060859, 2022.

Hamill, T. M.: Interpretation of Rank Histograms for Verifying Ensemble Forecasts, Monthly Weather Review, 129, 550–560, https://doi.org/10.1175/1520-0493(2001)129⟨0550:IORHFV⟩2.0.CO;2, 2001.

Hersbach, H.: Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems, Weather and Forecasting, 15, 559–570, https://doi.org/10.1175/1520-0434(2000)015⟨0559:DOTCRP⟩2.0.CO;2, 2000.

Hersbach, H., Bell, B., Berrisford, P., et al.: The ERA5 Global Reanalysis, Quarterly Journal of the Royal Meteorological Society, 146, 1999–2049, https://doi.org/10.1002/qj.3803, 2020.

Hess, P., Aich, M., Pan, B., and Boers, N.: Fast, Scale-Adaptive and Uncertainty-Aware Downscaling of Earth System Model Fields with Generative Machine Learning, Nature Machine Intelligence, 7, 363–373, https://doi.org/10.1038/s42256-025-00980-5, 2025.

Karras, T., Aittala, M., Laine, S., and Aila, T.: Elucidating the Design Space of Diffusion-Based Generative Models, in: Proceedings of the 36th International Conference on Neural Information Processing Systems, Nips '22, Curran Associates Inc., Red Hook, NY, USA, ISBN 978-1-7138-7108-8, 2022.

Lovejoy, S. and Schertzer, D.: The Weather and Climate: Emergent Laws and Multifractal Cascades, Cambridge University Press, https://doi.org/10.1017/CBO9781139093811, 2013.

LUMI: LUMI: Large Unified Modern Infrastructure, https://www.lumi-supercomputer.eu/, 2022.

Perez, E., Strub, F., de Vries, H., Dumoulin, V., and Courville, A. C.: FiLM: Visual Reasoning with a General Conditioning Layer, in: AAAI, 2018.

Perlin, K.: An Image Synthesizer, ACM SIGGRAPH Computer Graphics, 19, 287–296, https://doi.org/10.1145/325165.325247, 1985.

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S.: Deep Unsupervised Learning Using Nonequilibrium Thermodynamics, 2015.

Wu, R., Yang, L., Chen, C., Ahmad, S., Dascalu, S. M., and Harris Jr., F. C.: MELPF Version 1: Modeling Error Learning Based Post-Processor Framework for Hydrologic Models Accuracy Improvement, Geoscientific Model Development, 12, 4115–4131, https://doi.org/10.5194/gmd-12-4115-2019, 2019.

Yang, X., Hintz, K. S., Aros, C. P., and Amstrup, B.: Danish Atmospheric Reanalysis : Final Scientific Report of the 2020 National Centre for Climate Research Work Package 3.2.1, Regional Reanalysis Pilot, Tech. Rep. 9788774787051, Copenhagen, 2021.

Yang, X., Amstrup, B., Aros, C. P., and Hintz, K.: Danish Regional Reanalysis : Final Scientific Report of the 2021 National Center for Climate Research Work Package 1.1.2 Danish Regional Reanalysis, Tech. Rep. 9788774787150, Danish Meteorological Institute, Copenhagen, 2022.