Direct Sampling for Extreme Events Generation and Spatial Variability Enhancement of Weather Generators

Jorge Luis Guevara Diaz[1], Maria Garcia[1], Priscilla Avegliano[1], Debora Lima[1], Dilermando Queiroz[1], Maysa Macedo[1], Leonardo Tizzei[1], Daniela Szwarcman[1], Bianca Zadrozny[1], Campbell Watson[1], and Anne Jones[2]

[1]IBM Research
[2]IBM Research Europe

October 4, 2023

Direct Sampling for Spatially Variable Extreme Event Generation in Resampling-Based Stochastic Weather Generators

Jorge Guevara[1], Maria Garcia[1], Priscilla Avegliano[1], Debora Lima[1], Dilermando Queiroz[1], Maysa Macedo[1], Leonardo Tizzei[1], Daniela Szwarcman[1], Bianca Zadrozny[1], Campbell Watson[1], Anne Jones[1]

Corresponding author: Jorge Guevara(jorgegd@br.ibm.com)

[1]IBM Research

Key Points:

- Spatial variability improvement of weather generators based on resampling via the Direct Sampling algorithm
- Direct Sampling for extreme precipitation fields generation using control points and return periods
- Empirical validation using statistical and connectivity metrics on a dataset with precipitation, temperature and cloud cover variables

Corresponding author: Jorge Guevara, `jorgegd@br.ibm.com`

Abstract

Resampling-based weather generators simulate new time series of weather variables by reordering the observed values such that the statistics of the simulated data are consistent with the observed ones. These generators are fully data-driven, easy to implement, and capable of reproducing the dynamics among weather variables. However, although the simulated time series is new, the weather fields produced at arbitrary time steps are replicas of those found in observations, limiting the spatial variability of simulations and preventing the generation of extreme weather fields beyond the range of observed values. To address these limitations, we propose the integration of the Direct Sampling algorithm—a data-driven method for producing simulations—into resampling-based weather generators. By incorporating Direct Sampling as a post-processing step on the outputs of the weather generator, we enhance the spatial variability of the generated weather fields and enable the generation of extreme weather fields. We introduce an approach for generating out-of-sample extreme weather fields using Direct Sampling. This method involves utilizing a set of control points in conjunction with Direct Sampling, where the values of these control points are informed by return period analysis. The proposed approach is validated using precipitation, temperature, and cloud cover weather fields in a region of northwest India. The experimental results confirm that Direct Sampling enhances the spatial variability of the weather fields and facilitates the generation of out-of-sample precipitation fields that accurately adhere to the spatial statistics provided by return precipitation level maps, as well as the observed precipitation weather field employed in the analysis.

Plain Language Summary

Weather generators (WG) are tools for generating artificial weather data. Applications use WG outputs for several tasks, including risk, uncertainty, and climate change analysis. WGs based on "resampling " conforms to a type of WGs that is easy to implement, understand and produce data with properties resembling historical data. However, although those WGs generate new artificial time series, those series are sorted versions of historical weather fields (i.e., weather data values at the spatial domain) Furthermore, those WGs can't generate weather fields with out-of-sample data values, i.e., extreme weather. In this work, we research the applicability of the Direct Sampling algorithm for creating variations of the simulated weather fields by the WG, and for generating artificial precipitation fields with extreme values. We found that Direct Sampling post-processing of weather generator outputs is a simple approach to improve the variations of weather fields and for generating extreme precipitation fields conditioned on information provided by an extreme precipitation analysis. The methods exposed in our work show a way to improve the design of those WGs that can benefit several applications like the ones searching the generation of hypothetical extreme weather fields, or seeking better uncertainty quantification or estimates in risk analysis tasks.

1 Introduction

Stochastic weather generators are tools for generating synthetic data (a.k.a statistical simulations) of weather variables such as temperature, precipitation, humidity, atmospheric pressure, and wind speed at particular locations and also at daily, hourly or finer temporal scales. Those simulations need to be statistically equivalent to observed data – they should reflect the weather persistence, variability and reproduce the spatio-temporal dynamics and correlation structures among the different meteorological variables (Ailliot et al., 2015). Figure 1 illustrates how a weather generator works, they are trained using available weather data from the specific area of interest, and, optionally, they can be conditioned on projections and control variables. The output of the weather generator consists of time series data that exhibit statistical coherence with the weather

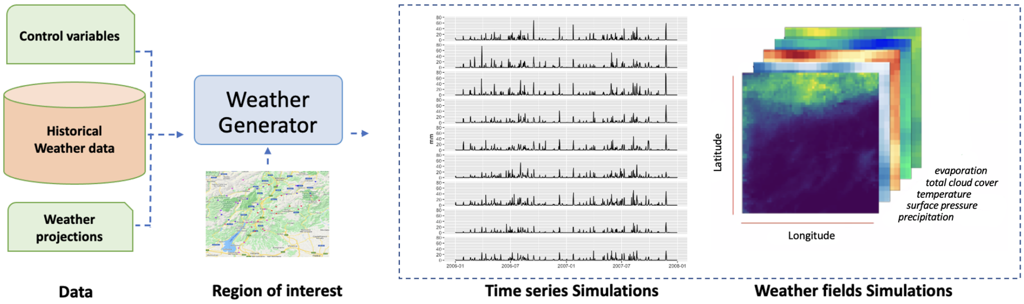

Figure 1. Weather generator pipeline: Weather generators use historical weather data to learn the proper underlying distribution to generate synthetic weather data statistically equivalent to observed weather data. Ideally, they can create samples conditioned on weather projections and alter their behavior due to the use of control variables.

data used in the training phase. They were first conceptualized by (Richardson, 1981) and have since become widely used to produce long surrogate time series and downscale future climate projections for climate impact assessments (Kilsby et al., 2007). For example, in hydrology, weather generators are used to create precipitation time series required to estimate flood risk or evaluate the sensitivity of the hydrological regime to climate change (Peleg et al., 2017). Other applications include future energy consumption impact analysis (Kolokotroni et al., 2012) and crop models (Brisson et al., 2009). Stochastic weather generators are low-cost computational tools — the data generation process is computationally cheap compared to climate models, which require solving complex physical equations (Peleg et al., 2017). Moreover, they are proper tools to explore uncertainty in climate (Peleg et al., 2019).

Stochastic weather generator types — Ailliot et al. (2015) considers there to be three types of stochastic weather generators: i) single-site refers to the ones that only synthesize data for a single weather station or location; ii) multisite are those that synthesize data for multiple stations or locations; iii) gridded are those weather generators that produce the so-called weather fields, essentially filling the gaps within the region of interest with simulated data. All three types can be univariate – e.g., they synthesize only precipitation values, or multivariate – they jointly synthesize several weather variables. Furthermore, by considering the methodological procedure behind their construction, (Ailliot et al., 2015) suggests four categories of weather generators: resampling methods, Box-Jenkins methodology, point process models, and hierarchical models. In this line, we can also group them into parametric, nonparametric, and semiparametric weather generators.

Parametric weather generators are those that rely on theoretical probability distributions to model the joint distribution of weather variables — for instance, precipitation is usually modeled by an exponential or gamma distribution (Todorovic & Woolhiser, 1975) and extreme rainfall by mixtures of gamma distributions (Kenabatho et al., 2012) or Generalized Pareto distribution (Lennartsson et al., 2008). Usually, the sampling process of synthetic weather data is conditioned upon a sequence of weather states sampled from a temporal occurrence model, and its variability within a region is modeled by a spatial model (Richardson, 1981; D. Wilks, 1998; Lee et al., 2010; Chen et al., 2012; Kim et al., 2012; Carey-Smith et al., 2014; Allard & Bourotte, 2015).

Nonparametric weather generators are fully data-driven and do not employ parametric probability distributions to specify the full joint probability distribution of weather variables. Popular methodologies for this include the use of empirical distributions (Semenov et al., 1998), neural networks (Trigo & Palutikof, 1999), and kernel density estimators (Rajagopalan et al., 1997). Further, semiparametric weather generators are those approaches based on parametric and nonparametric techniques for constructing weather generators. These methods improve persistence modeling issues found in nonparametric models (Apipattanavis et al., 2007; Steinschneider & Brown, 2013). They also include quantile mapping to enforce long-term distributional shifts in weather variables hypothesized by climate change scenarios, and explore more complex temporal occurrence and statistical models for improving weather generation (Steinschneider et al., 2019).

Weather generators based on resampling methods — Resampling methodologies are powerful and simpler strategies for generating synthetic weather data that capture the observed statistical properties of weather data like correlation structures among variables, weather persistence, and spatio-temporal variability (Rajagopalan & Lall, 1999; Buishand & Brandsma, 2001). They are usually used to construct nonparametric weather generators, or embedded within semiparametric weather generators. The resampling methods' main advantage is that the resampling process is performed jointly for all the variables and all the sites, which guarantees the spatial coherence of the synthetic weather data. So, as quoted by (D. S. Wilks & Wilby, 1999) — "they can capture deviations from theoretical probability distributions for the individual variables, and nonlinearities in the relationships among variables". In contrast, parametric weather generators have an inherent design complexity in assuming, choosing, and fitting parametric models within the model pipeline process, limiting their applicability to only small temporal and spatial scales (Wilcox et al., 2021).

Although weather generators based on resampling can generate new weather data time series, the spatial weather fields they produce are replicas or calibrated versions of the observed historical data. Hence, the scope of the spatial variability of the synthetic time series is limited to the resampling algorithm and the historical weather data. That is an issue for the synthetic weather fields generation task. Furthermore, it is impossible to generate hypothetical weather fields with extreme events without relying on scaling factors applied to the observed ones. Additionally, resampling methodologies strongly depend on the quantity and quality of available data.

1.1 Contributions

We propose to overcome the spatial variability and the out-sample extreme event generation limitations of weather generators based on resampling by using the Direct Sampling — which is a Multi-point Geostatistics algorithm capable of generating sets of simulations based on the patterns of a training image and conditioning data (Mariethoz et al., 2010) — as a postprocessing step on the weather generator outputs. We consider weather generators that produce new time series of weather fields, and they are multivariate weather generators, i.e., for a given time, they produce a set of weather fields, such that each weather field corresponds to a weather variable. Although the proposed methodology can work with any weather generator based on resampling, we showcase our experimental analysis using a weather generator implementation based on the works from (Rajagopalan & Lall, 1999; Apipattanavis et al., 2007; Steinschneider & Brown, 2013). Thus, the paper's contributions are:

- The use of the Direct Sampling algorithm to improve the spatial variability of multivariate weather generators based on resampling
- Generation of weather fields with extreme events from the weather generator outputs using Direct Sampling with control points informed by a return period analysis.
- Experimental validation using a set of statistical and connectivity metrics and a weather dataset with precipitation, temperature and cloud-cover weather fields for a region in northwest India

2 The weather generator and Direct Sampling algorithms

This section introduces the weather generator based on resampling used in the experiments and the Direct Sampling algorithm from Multi-point Geostatistics.

2.1 The weather generator

This work employs an implementation of the multisite and multivariable weather generator presented in (Steinschneider & Brown, 2013). The weather generator begins by estimating the area-averaged total annual precipitation for a target year, which serves as a reference for simulating weather values at various sites in a given region. Depending on the target year (historical or future year), the area-averaged total annual precipitation can be calculated using historical observations or through the use of forecaster models such as WARM (Kwon et al., 2007; Steinschneider & Brown, 2013) or ARIMA (via the *weathergen R* package). The estimated area-averaged total annual precipitation for the target year is then used to construct a bootstrapped sample, which contains the one hundred most similar years to the target year in terms of their area-averaged total annual precipitation. The bootstrapping algorithm employs the empirical distribution of weighted Euclidean distances between the estimated area-averaged total annual precipitation of the target year and the area-averaged total annual precipitation from the bootstrapped sample. The weights for the distances are computed using the kernel proposed by (Lall & Sharma, 1996).

The weather generator next step is to use the one hundred bootstrapped years to train twelve first-order homogeneous Markov chains with three states — dry, wet, and extreme — i.e., a Markov Chain per month. For that purpose, the weather generator previously labeled the area-averaged daily precipitation signal from the one hundred bootstrapped years with those three states based on user-provided threshold values, for instance, the label 'dry' is given to precipitation values less than 0.01 mm/day, 'wet' to values greater than 0.01 mm/day but less than the 90th percentile of monthly precipitation values, and 'extreme' otherwise. Thus, the weather generator simulates a state sequence for a year sampling from the twelve Markov chains. The weather generator's next stage is to estimate the resampling dates for the one-year state sequence. This is achieved using a 1-lag K-nearest neighbors (KNN) bootstrap algorithm. The algorithm assigns a date to a state s_t based on the empirical distribution of weighted distances between the weather values corresponding to the date already assigned to state s_{t-1} and all the weather values in the training dataset that correspond to the first state within a similar state sequence $s_{t'-1}, s_{t'}$, where t' is an index within an α-day window within the training data. Finally, the weather generator uses the resampling dates from the previous step to assign the weather values from the historical training data to the simulation.

2.2 Direct Sampling algorithm

Direct Sampling is an algorithm from Multi-point Geostatistics — an area that focuses on stochastic modeling based on training images instead of using traditional Random Function Theory (Mariethoz et al., 2010; Mariethoz & Caers, 2014). The main use of Direct Sampling is to provide statistically coherent simulations with a structure mimicking the one provided by a training image, whose main role in applications is to inform and include physical reality in stochastic modeling. Some Direct Sampling and Multipoint Geostatistics applications on the weather domain are on conditional stochastic rainfall (Wojcik et al., 2009), downscaling (Jha et al., 2013, 2015), resampling extremes (Opitz et al., 2021), rainfall series generation (Benoit & Mariethoz, 2017; Oriani et al., 2014, 2018) and conditional weather field generation (Oriani et al., 2017).

The Direct Sampling algorithm starts by using a user-provided *training image* as the source of patterns for creating simulation. The training image is represented by a two-dimensional matrix $\mathbf{W}$, where the matrix indices correspond directly to the spatial coordinates of the data. To construct a new image that is similar in structure to the training image, the algorithm uses a two-dimensional empty matrix $\mathbf{S}$ as a *simulation grid.* Importantly, the size of $\mathbf{S}$ does not need to be the same as that of $\mathbf{W}$, enabling the generation of images of varying sizes. Moreover, the algorithm can incorporate *conditioning data* by using a two-dimensional matrix $\hat{\mathbf{C}}$ to store user-provided values at specific locations that the simulation must obey. The conditioning data matrix $\hat{\mathbf{C}}$ is empty everywhere except for the locations with conditioning data. A *data event* for a location $\mathbf{x}$ in the simulation grid $\mathbf{S}$ is defined as the set of n locations around $\mathbf{x}$ where simulated point values have already been assigned We denote this set as $d_n(\mathbf{x}, L) = \{Z(\mathbf{x}+\mathbf{h}_1), \ldots, Z(\mathbf{x}+\mathbf{h}_n)\}$, where L represents a set of lag vectors $\{\mathbf{h}_1, \ldots, \mathbf{h_n}\}$ that define a neighborhood around $\mathbf{x}$, and $Z(\mathbf{x})$ is the simulated value for location $\mathbf{x}$. A data event for a location $\mathbf{y}$ within the training image $\mathbf{W}$ also uses the set of lag vectors L already computed from $\mathbf{S}$ but in $\mathbf{W}$, i.e., $d_n(\mathbf{y}, L) = \{Z(\mathbf{y} + \mathbf{h}_1), \ldots, Z(\mathbf{y} + \mathbf{h}_n)\}$. The Direct Sampling algorithm uses a distance function to compare the data events of the training image and the simulation grid, i.e., $D(d_n(\mathbf{x}, L), d_n(\mathbf{y}, L))$. Additionally, the algorithm employs a threshold value th on the distances as a criterion for assigning values to the simulation grid.

The Direct Sampling algorithm is executed in the following steps:

1. Define the necessary input matrices, $\mathbf{W}$ and $\hat{\mathbf{C}}$, and set the values of the parameters n, D, and the threshold th.
2. Initialize the simulation grid, $\mathbf{S}$, with the conditional data point values at the same locations provided by $\hat{\mathbf{C}}$, and set empty values elsewhere.
3. Randomly select a location $\mathbf{x}$ with empty value from $\mathbf{S}$, and compute the corresponding data event $d_n(\mathbf{x}, L)$ and the set of lag vectors L.
4. Randomly select a location $\mathbf{y}$ from $\mathbf{W}$, and compute the corresponding data event $d_n(\mathbf{y}, L)$.
5. If the distance between the two data events, $D(d_n(\mathbf{x}, L), d_n(\mathbf{y}, L))$, is less than or equal to the threshold th, assign the value of $\mathbf{y}$ from $\mathbf{W}$ to the location $\mathbf{x}$ in $\mathbf{S}$. If there are more locations $\mathbf{x}$ with empty values to explore in $\mathbf{S}$ proceed to step 3 otherwise the algorithm ends.
6. If $D(d_n(\mathbf{x}, L), d_n(\mathbf{y}, L))$ is greater than th, store the pair $\{D(d_n(\mathbf{x}, L), d_n(\mathbf{y}, L)), \mathbf{y}\}$ in a list. If there are more locations $\mathbf{y}$ to explore in $\mathbf{W}$, go to step 4. Otherwise, choose in the list the value of $\mathbf{y}$ with the shortest data event distance and assign it to the location $\mathbf{x}$ in $\mathbf{S}$.

An interesting property of this algorithm is that the pattern matching between the TI and SG data events looks for a very diverse set of structures at different scales — it scrutinizes for a myriad of patterns of different sizes without using predefined templates. Also, it allows the use of continuous and discrete variables, co-simulation, conditioning points, multiple variables, and parallel algorithms. Furthermore, the quality of the simulations depends on the quality of the TI and the parameter settings such as the distance between data events, threshold distance value, and data event size. All of that will require a first round of sensitivity analysis to calibrate the algorithm with the best set of parameters. Unfortunately, there is a positive correlation between the best parameter configuration and a high computational cost, however, some solutions based on GPU and parallel implementations exist (Huang et al., 2013; Cui et al., 2021). In this paper, we distributed Direct Sampling algorithm instances among several CPU cores working in parallel. We described this approach in the Supplemental Material.

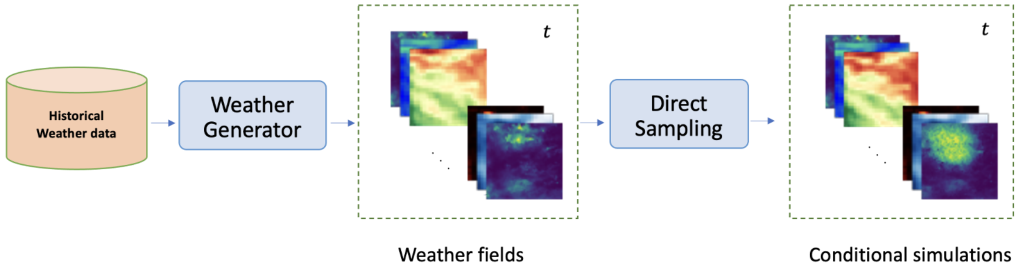

Figure 2. Post-processing the weather field outputs from the weather generator with the direct sampling algorithm for spatial variability improvement.

3 Using Direct Sampling within the weather generator

This section proposes a strategy to couple the Direct Sampling algorithm within weather generators based on resampling for two tasks: 1) spatial variability improvement and 2) extreme weather scenarios generation.

3.1 Improving spatial variability

Resampling-based weather generators are limited in their ability to generate weather fields with spatial variability that differs from the historical training set. To overcome this issue, we propose to use the Direct Sampling algorithm as a post-processing step of the weather generator outputs to produce new weather field variations and hence increase the weather generator spatial variability. In this sense, the Direct Sampling procedure considers each weather field produced by the weather generator at time t as a training image. An important aspect to take into account is that the Direct Sampling post-processing procedure needs to consistently keep the original statistical properties and connectivity structures of the original weather fields. Furthermore, the post-processed weather field's time series must keep the correlated interplay dynamics among weather variables and coherently retain the temporal aspect modeled by the weather generator. This aspect is easily solved by using an appropriate set of conditioning points through the time and space dimensions to constraint the Direct Sampling outputs.

Figure 2 shows the Direct Sampling post-processing of the weather fields produced from the weather generator at time t. The Direct Sampling procedure uses a unique set of randomly located conditional points within the spatial domain in analysis. The locations are shared for all the weather fields and all $t = 1, \ldots, T$, but the values of the conditional data are taken from each respective training image (i.e., each synthetic weather field produced from the weather generator). The reason behind this is to guarantee statistical coherence among all the simulated weather fields variables by Direct Sampling at time t. It should be noted that each weather field generated by the weather generator is not only used as a training image, but also as a source of conditioning data to restrict simulations to adhere to specified values or patterns at selected locations. This approach allows for the implicit integration of the effects of topography to some extent. Also, instead of using a multivariable version of Direct Sampling, i.e. simulating several variables at once, we opt for the univariate version, we chose this approach due to its simplicity. Formally, let $\mathbf{W}$ be a matrix representing a weather field produced by the weather generator. Let $\mathbf{S}$ be the matrix representing the simulation grid. Also, let I be an index set with random locations in the spatial domain of interest. Thus for a weather field $\mathbf{W}$ produced by the weather generator for a weather variable $v \in \mathcal{V}$ at time t, construct a conditional data matrix $\hat{\mathbf{C}}$, such that $\{\hat{\mathbf{C}}_i = \mathbf{W}_i\}_{i \in I}$ and $\{\hat{\mathbf{C}}_i = \phi\}_{i \notin I}$, where ϕ de-

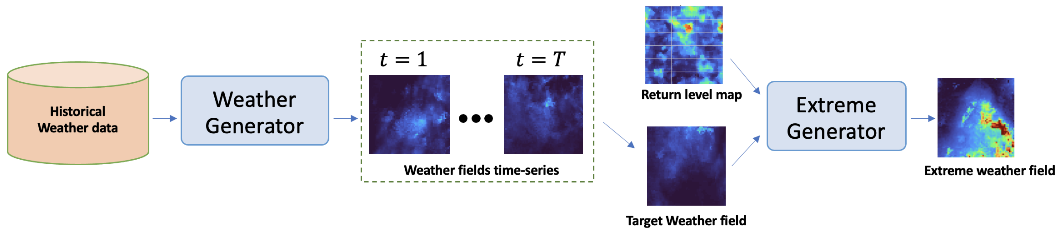

Figure 3. The extreme precipitation weather field generation pipeline: the weather generator produces a statistically consistent time series of precipitation weather fields, then, an extreme generation procedure produces an extreme precipitation weather field from a simulated target weather field.

notes the null element. Using that information, run the Direct Sampling algorithm considering $\mathbf{W}$ as the training image, and conditional data matrix $\hat{\mathbf{C}}$ to produce a simulation $\mathbf{S}$. Repeat the procedure for all the weather fields per weather variable $v \in \mathcal{V}$ (i.e., precipitation, temperature, humidity, etc.) and all times t. Observe that, in hypothesis, the new weather field $\mathbf{S}$ provides a variation with statistical and structural properties similar to $\mathbf{W}$.

3.2 Generation of weather fields with extreme events

An important issue of weather generators based on resampling is the inability to generate extreme weather fields outside the range of historical extremes In the case of extreme precipitation, we propose a method to generate out-of-sample extreme precipitation weather fields. This approach involves using Direct Sampling on a target weather field, conditioned on a return level map and control points. Notably, this procedure eliminates the need for quantile mapping and offers greater flexibility in producing extreme precipitation events within user-defined regions of interest. In Figure 3, we present a pipeline depicting the generation of weather fields with extreme precipitation values. The extreme generator selects a target weather field from the simulations provided by the weather generator and applies the aforementioned procedure to produce extreme precipitation values, taking into account the information provided by a return precipitation level map. To ensure completeness, we also generate extreme precipitations using quantile mappings on Direct Sampling simulations, which are described in Supplemental Material.

3.2.1 Direct Sampling on a target weather field conditioned on a return level map and control points

Generating extreme precipitation weather values in arbitrary areas of interest can provide valuable information for downstream applications, such as flood risk analysis, streamflow models, disaster management, and risk assessment. To achieve this, stakeholders can define the regions of interest in advance and use simulated extreme precipitation values correlated with the return period analysis to inform downstream applications. This method could enable stakeholders to better understand and manage the potential impact of extreme weather events on their communities, enhancing their resilience to future natural disasters. One approach to generate extreme weather fields is to condition the Direct Sampling on a set of control points defining the locations for generating extreme weather values. In this method, the control points' values are defined from the return level map values associated with a given return period. This approach avoids the need for quantile mapping and the assumption of parametric distributions, providing the additional benefit of using control points and return level maps to condition the generation of extreme precipitation weather values in arbitrary areas of interest. We propose the following procedure:

1. Identify a target weather field $\mathbf{W}'$ in $\mathbf{W}_t, (t = 1, \ldots, T)$.
2. Set the weather field $\mathbf{W}'$ as a training image.
3. Set the conditioning data matrix $\hat{\mathbf{C}}$ to the weather values of random locations within the region of interest that we do not want to generate extreme precipitation, i.e., $\{\hat{\mathbf{C}}_i = \mathbf{W}'_i\}_{i \in \hat{I}}$ and $\{\hat{\mathbf{C}}_i = \phi\}_{i \notin \hat{I}}$, where $\hat{I}$ is an index set of random locations in the spatial domain of interest.
4. Set the control points data matrix $\mathbf{C}$ to (out-of-sample) extreme weather values at locations where we want to generate extreme precipitation events. i.e., $\{\mathbf{C}_i = f_i\}_{i \in I}$ and $\{\mathbf{C}_i = \phi\}_{i \notin I}$, where f is a random process depending on a provided return precipitation level map $\mathbf{M}$ and a specific location, that is, $f_i = f(i, \mathbf{M})$ and I is an index set of locations in the spatial domain of interest where we want to generate extreme weather. We call all the points $\mathbf{C}_i \neq \phi$ as *control points.*
5. Assign to the simulation grid $\mathbf{S}$ all the points in the conditioning and the control points data matrices, i.e, $\{\mathbf{S}_i = \hat{\mathbf{C}}_i\}_{i \in \hat{I}}$ and $\{\mathbf{S}_i = \mathbf{C}_i\}_{i \in I}$
6. Run the Direct Sampling algorithm with Training Image $\mathbf{W}'$ and Simulation grid $\mathbf{S}$, and appropriate parameters as usual but if in the process of simulating a location $\mathbf{x}$ in the grid $\mathbf{S}$, a value is found in the data event belonging to the simulation grid greater than the maximum value found in the training image, i.e., $\max d_n(\mathbf{x}, L) > \max \mathbf{W}'$, do the following update: $\mathbf{S}_\mathbf{x} = \mathbf{W}'_\mathbf{y} - \overline{d_n(\mathbf{y}, L)} + \overline{d_n(\mathbf{x}, L)}$, where $\mathbf{W}'_\mathbf{y}$ is a random point in the training image located at $\mathbf{y}$, the overline notation denotes de average and $d_n(\mathbf{y}, L)$ and $d_n(\mathbf{x}, L)$ are the data events of the training image $\mathbf{W}'$ and simulation grid $\mathbf{S}$, respectively.

Observe that the main purpose of conditioning data matrix $\hat{\mathbf{C}}$ is to guarantee spatial coherence and honor some values of the training image data. Additionally, this conditioning enables the implicit incorporation of topographical effects to a certain extent within the simulations. Also, the locations in $\hat{\mathbf{C}}$ and $\mathbf{C}$ can be selected by users or by analyzing some precipitation statistics within the region of interest. The update for $S_\mathbf{x}$ defined in step 6, was mentioned in (Mariethoz et al., 2010) in the context of non-stationary distances. Figure 4 depicts the proposed pipeline using a 100-year return precipitation level map and a target precipitation weather field (training image) from the weather generator. The region of interest is the rectangular yellow area, which defines the location of the control points where Direct Sampling will generate extreme precipitation values. On another hand, the conditioning points are randomly located outside the region of interest (yellow area) with values taken from the training image. The generated precipitation weather field with extreme values is shown on the right.

3.2.2 Potential applications of synthetic extreme weather generation

Extreme rainfall events can have devastating consequences on communities and infrastructure, particularly in areas prone to flooding. The use of extreme rainfall generation can simulate the effects of heavy precipitation and assist in developing effective mitigation strategies. In hydrological modeling, for instance, extreme rainfall generation can be used to predict the potential impacts of extreme precipitation events on streamflow (Oliveira et al., 2021), water quality (Exum et al., 2018), and ecosystem health (Wang et al., 2018). It can also be used to improve the design of water resources systems, such as reservoirs and irrigation systems, by simulating the effects of extreme rainfall on the systems' performance (Samuels et al., 2009; Woldemichael et al., 2012). Additionally, extreme rainfall generation can be applied in disaster risk reduction planning by identifying vulnerable areas and developing effective mitigation strategies (Revi, 2008). In the field of urban planning, the use of extreme rainfall generation can assist in designing resilient stormwater management systems, improving drainage infrastructure, and

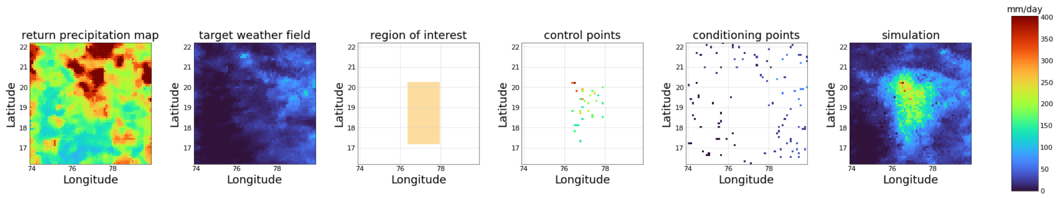

Figure 4. Example workflow for generation of an extreme precipitation event using Direct Sampling conditioned on a return precipitation level map via control points. The algorithm selects a target weather field as a training image and receives two external inputs: a return precipitation level map and a set of control points within a region of interest. The control points couple the locations in the target weather field with values from the return precipitation level map. Further, a set of conditioning points are used as a way to hold the non-stationarity and connectivity properties from the original target weather field. Direct Sampling uses such information to generate a simulation of an extreme precipitation weather field.

enhancing the overall resilience of urban areas to extreme weather events (Zhou et al., 2013).

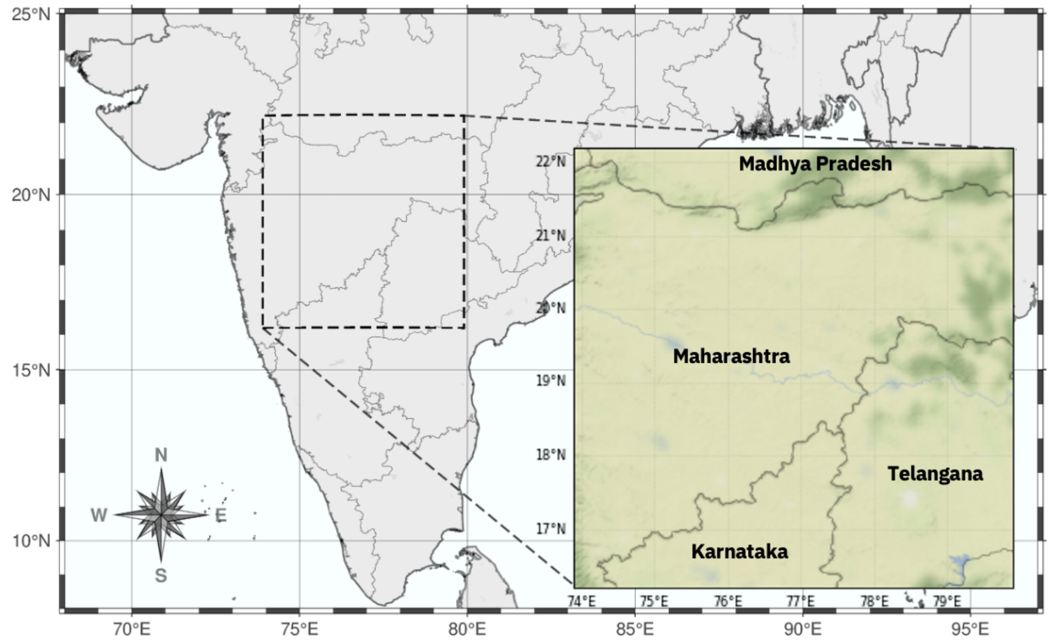

Figure 5. Map of India with case study highlighted.

4 Experimental Evaluation

4.1 Dataset and region of interest

The case study chosen is a region in northwest India, located at 16.2° N, 73.9° E, 22.2° N, 79.9° E, which contains portions of Maharashtra, Telangana, Madhya Pradesh, and Karnataka states (see Figure 5). The total area comprises around 360 000 km^2 and contains 3 600 latitude and longitude pairs. In this work, we use the daily *IMERG* dataset (Huffman et al., 2019) and the *ERA5* (single levels) (Hersbach et al., 2018). The IMERG-F V06 (IMERG Final Precipitation) is a run intended for research, which provides global precipitation estimates at the daily interval and at 0.1° spatial resolution (0.1° × 0.1°, corresponding to approximately to 10km × 10km). The ERA5 (single levels) is a fifth-generation ECMWF reanalysis product with a global range, which provides many atmospheric variables at the hourly interval and 0.25° spatial resolution (atmosphere). The weather variables we used were precipitation from IMERG-F, air temperature measured at 2 meters above the ground from ERA5, and total cloud cover data from ERA5. To align the datasets in temporal and spatial dimensions, we aggregated the ERA5 hourly temperature and cloud cover to daily values by adding them up and performed a nearest neighbor interpolation for downscaling from 0.25° × 0.25° to 0.1° × 0.1°. The final dataset covers 21 years (2001-2021) with three daily weather variables: precipitation, temperature, and total cloud cover.

4.2 Metrics

We used a series of quantitative metrics to validate how well the simulations are reproducing the structural and statistical properties of the training images. We employed the following metrics to quantify the reproduction of statistical properties:

- Quantile-quantile plot between the pixel values of the training image and the simulation

- Comparison between the empirical cumulative distribution functions (eCDF) between the pixel values of the training image and the simulation
- Comparison between variograms (denoted by γ) estimated from the training image and the simulation.

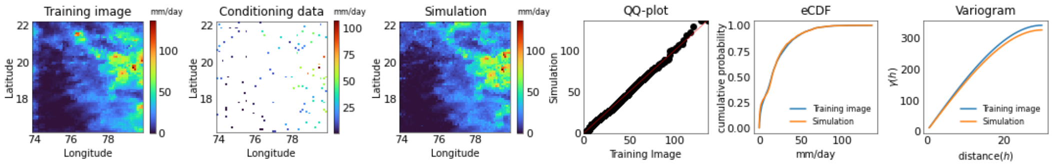

Figure 6. The quantile-quantile, eCDF, and Variogram metrics show how the simulation preserves some statistical properties of the training image: the distribution of pixel values and the variation of the spatial dependence.

Figure 6 shows from left to right a training image (a precipitation weather field $\mathbf{W}$ produced by the weather generator), the conditioning points $\mathbf{C}$, the simulation grid $\mathbf{S}$ and the quantile-quantile, eCDF, and Variogram plots. For this particular simulation, those metrics agree that the simulation preserves the statistical properties of the training image's pixel values.

We used the following metrics to quantify the reproduction of connectivity properties:

- **Two-point probability function** (Renard & Allard, 2013; Torquato et al., 1988; Torquato & Haslach Jr, 2002) — This function assumes that the input is a binary image $\mathcal{I}$, which is a matrix with zeros and ones. It measures the probability that two pixels located at $\mathbf{x}$ and $\mathbf{x}$+$\mathbf{h}$ contain the value one, given a lag vector $\mathbf{h}$. That is, $S_2(\mathbf{h}) = P\{\mathcal{I}(\mathbf{x}) = 1, \mathcal{I}(\mathbf{x}+\mathbf{h}) = 1\}$. If $||\mathbf{h}|| = 0$, then $S_2(\mathbf{h})$ is simply the probability that a pixel contains the value one, which equals the fraction of pixels in the image that have the value one, i.e., $S_2(\mathbf{h}) = E\{\mathcal{I}(\mathbf{x} = 1)\} = \varphi$. On the other hand, if the distance between the two pixels located at $\mathbf{x}$ and $\mathbf{x}$+$\mathbf{h}$ is very large (i.e., $||\mathbf{h}|| \to \infty$), then they become statistically independent. In this case, the two pixels have the same probability φ of containing the value one, thus $S_2(\mathbf{h}) = \varphi^2$.
- **Two-Point Connectivity Function** (Renard & Allard, 2013; Torquato et al., 1988; Torquato & Haslach Jr, 2002) — The two-point connectivity function is designed to analyze images consisting of clusters of connected pixels. Such images are represented as matrices, with positive integer values indicating cluster regions, and zero values denoting areas without a cluster. Using a lag vector $\mathbf{h}$, the two-point connectivity function determines the probability that two pixels at $\mathbf{x}$ and $\mathbf{x}+\mathbf{h}$ belong to the same cluster,that is $C_2(\mathbf{h}) = P\{C(\mathbf{x}) = C(\mathbf{x}+\mathbf{h}) \neq 0\}$. If $||\mathbf{h}|| = 0$, then $C_2(\mathbf{h})$ is equivalent to the probability of pixels in the image within a cluster, represented by φ. Conversely, when the distance between the two pixels is large (i.e., $||\mathbf{h}|| \to \infty$), the two pixels are not connected and $C_2(\mathbf{h}) = 0$.

Figure 7, shows the two-point probability and connectivity functions in the first and second row respectively, for the training image and the simulation from Figure 6. In the case of the two-point probability function S_2, the input to the procedure is two binary images, one for the training image (Binary TI) and another one for the simulation (Binary Sim), where all the points of interest are labeled as one (yellow area) and zero otherwise. We constructed those binary images to analyze if the simulation is reproducing the connectivity properties of the training image in regions with high precip-

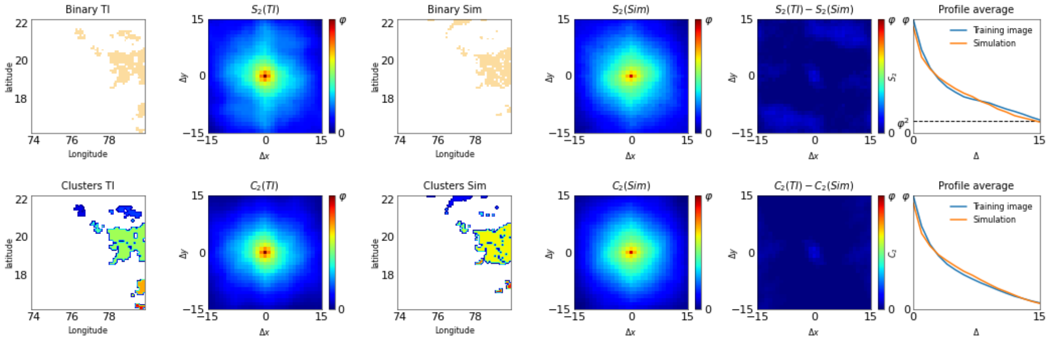

Figure 7. Probability and connectivity metrics for the training image and a simulation. The first row, from left to right, displays a binary image of the training image (Binary TI) and its corresponding probability function $S_2(TI)$, as well as a binary image from the simulation (Binary Sim) and its probability function $S_2(Sim)$. These binary images mask areas with high precipitation values, and the difference between $S_2(TI)$ and $S_2(Sim)$ is presented, alongside the profile average of the S_2 values for both training and simulation images in the X-axis, Y-axis, XY-axis, and YX-axis. The second row, from left to right, displays the clusters of high precipitation values for the training image (clusters TI) and its connectivity function $C_2(TI)$, as well as the cluster image for the simulation (cluster Sim) and its connectivity function $C_2(Sim)$. In addition, the difference between $C_2(TI)$ and $C_2(Sim)$ is shown, along with the profile average of the C_2 values. Notably, this example demonstrates a high level of agreement between the probability and connectivity metrics for both the simulation and training images.

itation values, thus, the yellow areas correspond to the places where the precipitation values exceed the 90th percentile of pixel values. With such information, we estimated the two-probability functions: $S_2(TI)$ and $S_2(Sim)$, for the training image and the simulation, respectively. Notation Δx and Δy refers to the components of the lag vector $\mathbf{h}$. In this case, those components vary from 0 to 15 pixels, which corresponds to 0.1° (10km approximately). We also show the difference : $S_2(TI) - S_2(Sim)$. Furthermore, we show for completeness the profile average of each two-probability function, which we computed by selecting the S_2 values in the X-axis, Y-axis, XY-axis, and YX-axis, starting from the center coordinates, and averaging it out. The x-axis label Δ refers to the common variation in directions X-axis, Y-axis, XY-axis, and YX-axis.

The input for the connectivity function C_2 shown in the second row of Figure 7 is two binary images describing the region of interest for the training image and the simulation likewise we mentioned before. The region of interest of both binary images is clusterized (Cluster TI and Cluster Sim). Using that information we estimated the C_2 function for the training image ($C_2(TI)$) and the simulation ($S_2(Sim)$). The figure also shows the difference image: $C_2(TI) - S_2(Sim)$ and the profile average.

In this paper, we utilize the two-point probability and connectivity functions to assess the fidelity of simulations in reproducing the connectivity properties across different precipitation levels. Specifically, we examine high precipitation values (greater than the 90th percentile of pixel values), middle precipitation values (pixel values between the 10th and 90th percentiles), and lower precipitation values (below the 10th percentile of pixel values) observed in the data. As an illustration, Figure 8 presents the metrics obtained from one hundred simulations compared to the training image depicted in Figure 6. The first row shows the training image, the mean of the simulations, and the standard deviation image of the simulations. The second row displays the QQ-plot, eCDF, and variogram of simulations (gray lines) and the training image (blue line). The third row shows the profile averages of the connectivity metric for the lower, middle, and higher precipitation values. Finally, the fifth row shows the probability metrics for the lower, middle, and higher precipitation values.

In the case of measuring the quality of simulation weather fields with extreme precipitation values outside the range of the training image, a reference image is imperative for comparison purposes. However it is extreme difficult to have a reference weather field to make comparisons. Even in the case of a reference weather field is available, it must be coherent with the training image used, as each extreme weather field generated is a function of the training image and the return level map. In such scenarios, relying on metrics is useful for informing the shift of precipitation value distribution and the variation of connectivity properties. For instance, higher values can be expected within the region of interest, while the connectivity properties should be maintained in regions with low precipitation values outside the region of interest. Therefore, while it may be difficult to obtain a reference weather field, utilizing appropriate metrics can provide meaningful insights in the process of extreme event generation.

4.3 Spatial variability enhancement of simulated daily weather fields

This section shows the experimental validation of using Direct Sampling to improve the spatial variability of weather generators based on resampling as we described in Section 3.1. As the quality of Direct Sampling simulations depend on its parameter choices we did a sensitivity analysis using precipitation, temperature and cloud cover weather fields from the data set described in Section 4.1.

4.3.1 Sensitivity analysis

The quality of Direct Sampling simulations depend on several parameter choices: 1) the distance function D between data events with threshold th to assign a value to the simulation grid — the lower the threshold, the higher the quality of the simulation; 2) the number of neighbors n that define the data events size — higher values could lead to poor simulation quality because the pattern in the simulation grid could not have a similar pattern in the training image (in this experiment is particularly true because we assume that the simulation grid is always the same size as the training image). On another hand, small values for n could lead to a narrow coverage of the structure of the patterns that the Direct Sampling is looking for.; 3) the number of conditioning points — in this experiment we set the conditioning points to be random locations from the training image, thus higher values could produce simulations looking very close to the training image and lower values would break such similarity; 4) a parameter f that describes the fraction of the training image that Direct Sampling uses for performing the pattern matching between the data events — a value close to 1 implies more computational cost but more pattern matching evaluations between the data events. On the contrary, a value close to 0 decreases the computational time at the expense of the simulation quality. To avoid an extensive parameter search, in the following experiment we fixed $f = 1$, i.e., we used the whole training image for the pattern matching search between the data events. While increasing the parameter f to a value of 1 and decreasing th towards 0 can lead to a higher probability of copying data from the same location of the training image into the same location of the simulation grid, resulting in lower variability, the use of control points as conditioning data, which varies for each realization, serves as a means to reintroduce variability. This approach allows us to maintain correlations among weather variables at specific times in the simulations and consider the effects of topography, all while balancing the need for variability in our results. We used a euclidean distance function for D with small threshold value of $th = 0.001$. Observe that with a small value for th we tradeoff computational time vs quality, since small values for th increase the quality of the simulations but increment the computational cost. Furthermore, we fixed

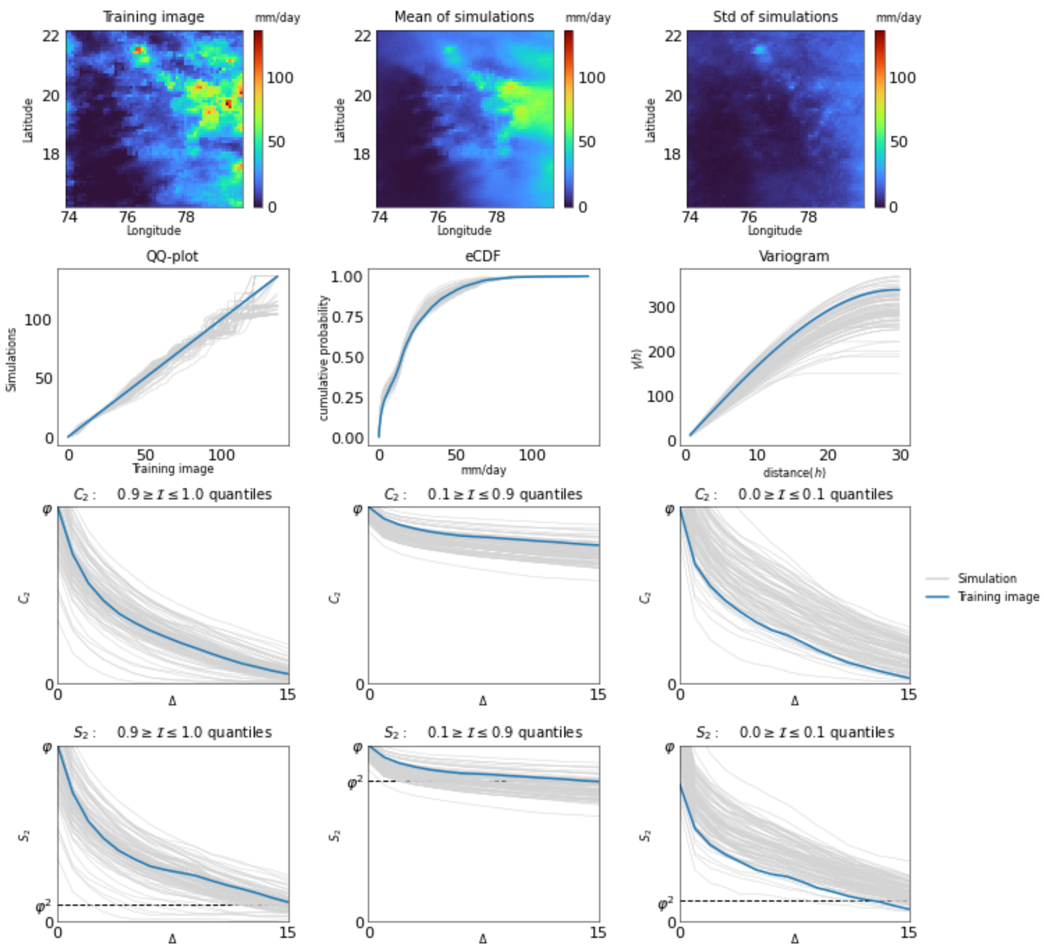

Figure 8. The statistical and connectivity metrics validate the quality of the simulations. The figure depicts the average and standard deviation images of one hundred simulations and a comparison between the simulation metrics and the training image metrics. The first row presents the training image, along with the mean and standard deviation images of the simulations. In the second row, the QQ-plot, eCDF, and variogram of the simulations (gray lines) are contrasted against the training image (blue line). The third row displays the profile averages of the connectivity metric, computed for the lower, middle, and higher precipitation values. Finally, the fifth row illustrates the probability metrics for these same precipitation levels.

the value of conditioning points to one hundred which roughly corresponds to 3% of the training image locations. Therefore, we constrain the sensitivity analysis only to the values of the n parameter.

For this experiment, we randomly selected one hundred precipitation, temperature, and cloud cover weather fields from the dataset satisfying the criteria that they do not have constant information, i.e., avoiding dry days in precipitation or no clouds in cloud cover weather fields — we set such weather fields as training images. Then, we used Direct Sampling to produce one hundred simulations per weather field, totalizing ten thousand simulations per each of the three weather variables. For each training image, we computed the curves produced by the metrics: QQ-plot, eCDF, Variogram, S_2 and C_2 and we did the same for its respective one hundred simulations. For instance, Figure 8,

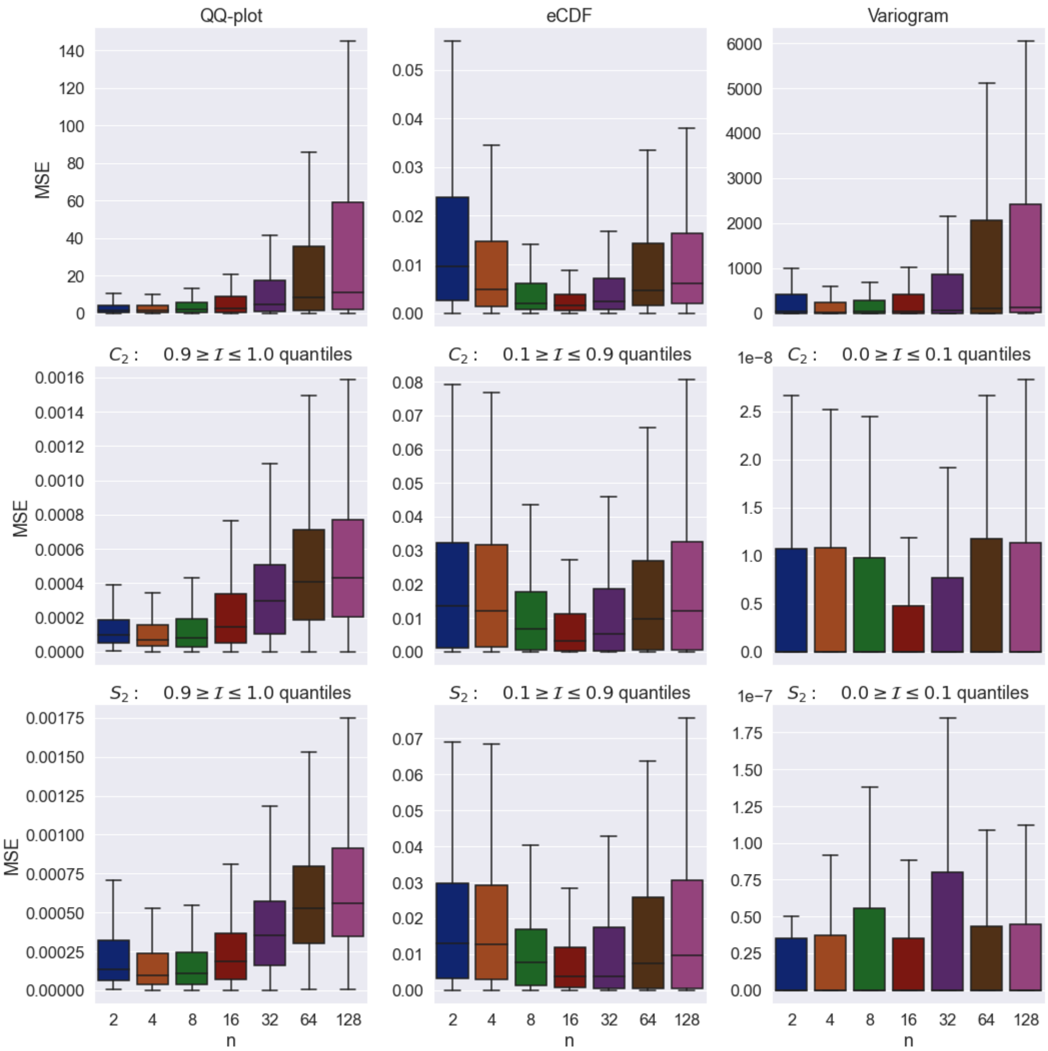

Figure 9. Sensitivity analysis for the Direct Sampling parameter N using several metrics for simulated precipitation weather fields. Each box plot represents a distribution of mean squared errors between the simulations and the training image metrics.

shows the case of one precipitation weather field that we used as a training image, the mean and the standard deviation images from the one hundred simulations, the curves produced by the metrics: QQ-plot, eCDF, Variogram, and the profile averages for the connectivity metrics at three different quantile intervals.

Figure 9 shows the results of a sensitivity analysis of Direct Sampling simulations of precipitation weather fields as a function of the n parameter (x-axis) and the metrics discussed so far. Each box plot represents the empirical distribution of the mean square errors (MSEs) computed between the curves (produced by the metrics) of the simulations and the training images. Thus, each box plot represents the empirical distribution of ten thousand MSE values. The MSE for the connectivity metrics S_2 and C_2 was estimated between the pixel values from images $S_2(TI)$ and $S_2(Sim)$, and $C_2(TI)$ and $C_2(Sim)$ (see Figure 7). We observed that the best value n in common for all the statistical metrics (QQ-plot, eCDF and Variogram) is when $n = 16$ or $n = 8$. In the case of the S_2 and C_2 connectivity metrics, the structure and connectivity properties of low precipi-

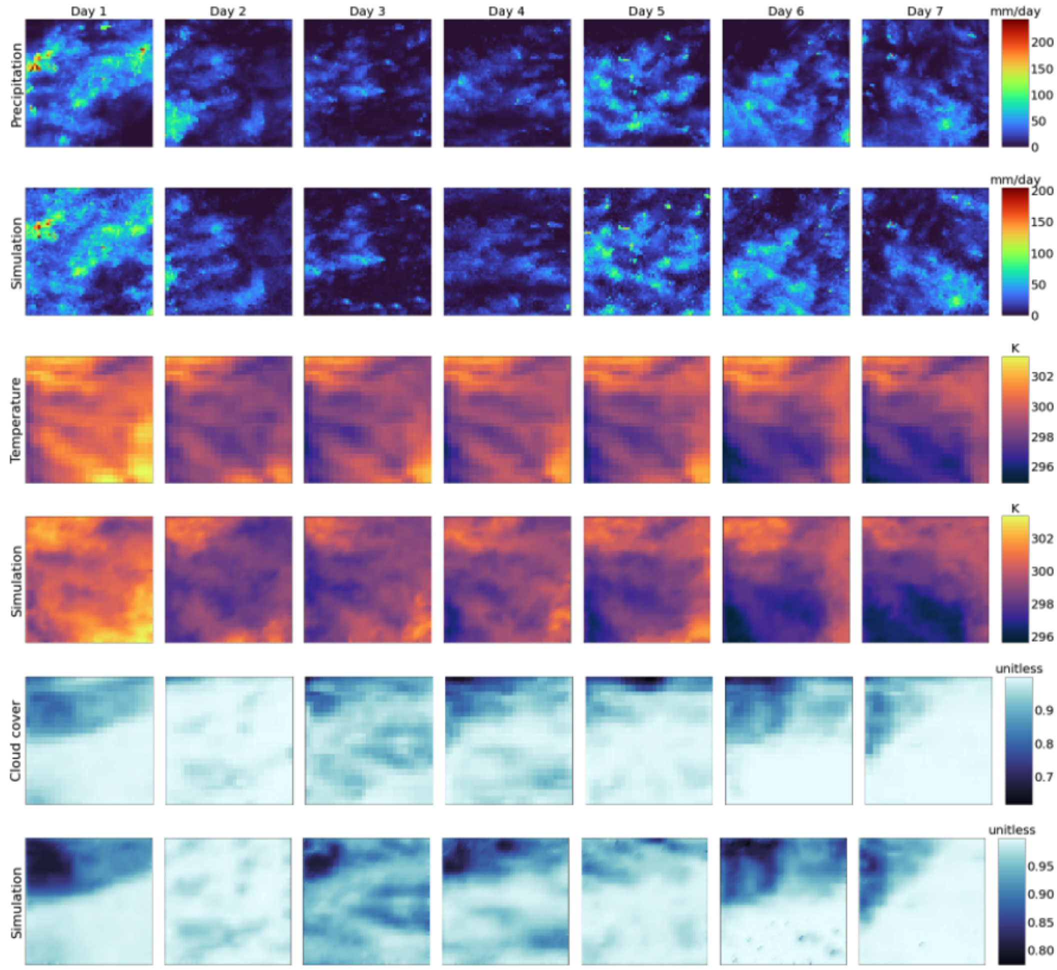

Figure 10. Precipitation, temperature, and cloud cover time series produced by the weather generator and the post-processed weather fields by the Direct Sampling algorithm labeled as *Simulation*. Temperature and cloud cover simulations are denoised because their original training images are pixelated and tend to generate artifacts

tation values — precipitation below the 10th percentile of the distribution of pixel values of the training image — is well reproduced by the simulations because the very small MSE values. In this sense, a value of $n = 16$ is good for reproducing the structure and connectivity properties of middle precipitation values (above the 10th percentile but less than the 90th percentile of the distribution of pixel values of the training image), and $n = 8$ or $n = 4$ for reproducing the structure and connectivity properties of high precipitation values (above the 90th percentile of the distribution of pixel values of the training image). A similar sensitivity analysis reveals that good choices for temperature are when $n = 4$ and for cloud cover $n = 16$.

4.3.2 Results on daily precipitation, temperature and cloud cover generation

In this experiment, we used the weather generator to simulate precipitation, temperature, and cloud-cover weather field time series. Then, we applied Direct Sampling on those weather fields to generate variations of those weather variables. Figure 10 shows

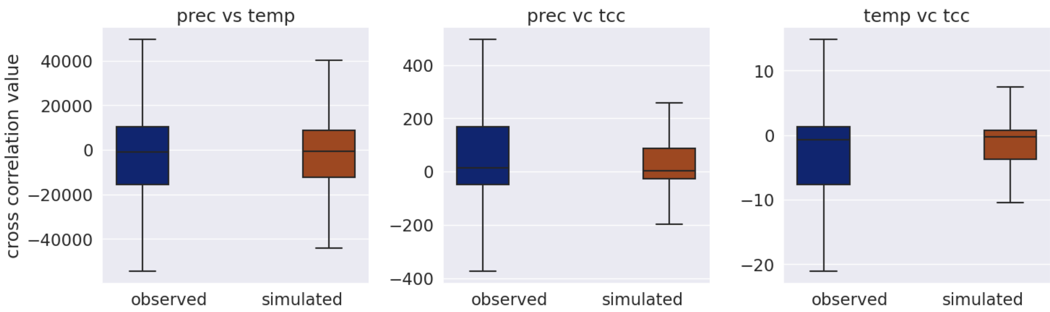

Figure 11. Cross-correlation results between each pair of variables.

seven consecutive days of weather fields produced by the weather generator. The rows labeled as *precipitation*, *temperature* and *cloud-cover* show the time series of weather fields produced by the weather generator, which are the training images used by Direct Sampling. The rows labeled as *simulation* show the simulations produced by the Direct Sampling. Based on the sensitivity analysis, we used $N = 16$ for precipitation and cloud-cover and $N = 4$ for temperature with one hundred randomly located conditioning points, such that the same locations are shared among all the weather variables at time t. Moreover, the temperature and cloud-cover weather fields are low-quality pixelated images with dominant low-frequency components. Therefore, Direct Sampling will produce simulations with some pixelation and noise artifacts. We applied a denoising process as suggested in the Direct Sampling literature (Meerschman et al., 2013). In this case, we used the information of the four neighbors around each pixel to determine if that pixel is a potential noise. Simulations results in Figure 10 shows the denoised results.

We estimated cross-correlation maps between each pair of weather field variables: precipitation vs. temperature, precipitation vs. total cloud cover, and temperature vs. total cloud cover, to measure how well the simulations honored the multivariate dependency among the variables. To this end, we sampled one hundred days from the historical data and selected the respective precipitation, temperature, and total cloud cover weather fields. For each day, we computed a cross-correlation map for each pair of observed weather field variables using $\mathbf{G}_{i,j} = \sum_{u=-k}^{k}\sum_{v=-k}^{k}\mathbf{H}_{u,v}\mathbf{W}_{i+u,j+v}$, where the pair $\mathbf{H}$ and $\mathbf{W}$ represent the combinations precipitation vs. temperature, or precipitation vs. total cloud cover, or temperature vs. total cloud cover. Also, the $\mathbf{H}$ and $\mathbf{W}$ are the centering matrix versions of the original weather fields. Figure 11 shows the distribution of the values of the cross-correlation maps for the observed weather fields as a blue box-plot. We also computed cross-correlation maps from the simulations provided by the Direct Sampling. In this case, for each day, we generated one hundred simulations of the three weather variables, and we computed the correlation maps as before. Figure 11 shows the distribution of the values of the cross-correlation maps for the simulations. Each box plot in this case has the distribution of ten thousand cross-correlation map values. From the results we observed that the distribution of the cross-correlation values for precipitation vs temperature is quite similar, and for the case of precipitation vs total cloud cover and temperature vs total cloud cover are within the expected range of the distribution of the observed ones.

4.4 Generation of precipitation weather fields with extreme events

In this experiment, we generated precipitation weather fields with extreme events using the approach described in Section 3.2. For this, we used the IMERG precipitation dataset as described in 4.1 to train the weather generator. We also estimated the return precipitation level maps for 100, 250, and 500-year return period events. Figure 12 shows

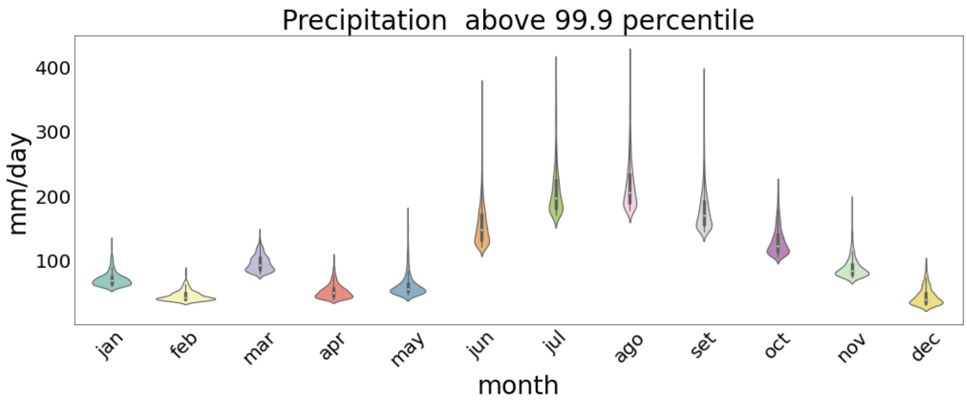

Figure 12. Distribution of the extreme daily precipitation grouped by month in the region under analysis, for the period 2001-2020.

the empirical distribution of daily precipitation values above the 99.9th percentile and grouped by month for the period 2001-2020 — observe that historical extreme precipitation values surpass the 400 mm/day for July and August months, which correspond to the Moonson period in central India.

4.4.1 Return precipitation level map estimation

The return period is a common risk measure used in the climate domain to assess the probability of extreme events and potential failures (Brunner et al., 2016; Vogel & Castellarin, 2017). However, empirical estimation of events with long return periods can be difficult due to limited data. To address this, we utilized Extreme Value Theory (De Haan & Ferreira, 2007) to generate return level maps for 100, 250, and 500-year return periods based on only 20 years of data. Our approach used a block-maxima sampling method collecting the most extreme precipitation events of each year and then using these samples to calibrate the parameters of a Generalized Extreme Value distribution (GEV) with the Maximum Likelihood method . The resulting return precipitation level maps are presented in Figure 13.

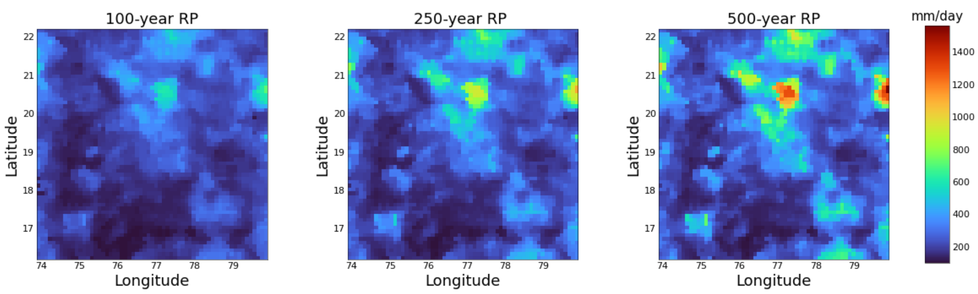

Figure 13. Return precipitation level maps for 100, 250 and 500 years of return periods.

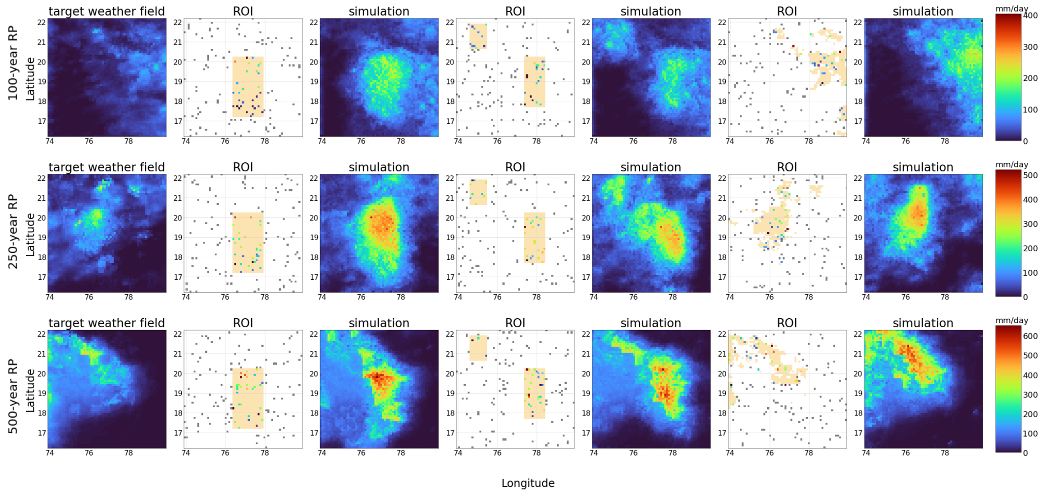

Figure 14. Extreme precipitation generation using the Direct Sampling and control points located in the yellow area of the ROI, conditioned on a 100, 250 and 500-year return precipitation level maps. Each control point's value is uniformly sampled from an interval that is defined by the maximum precipitation value in the image and the value at the location of the control point on the return precipitation level map

4.4.2 Conditioning the Direct Sampling on a return precipitation level map and control points

In this section, we show the use of the Direct Sampling conditioned on control points and return level maps as described in Section 3.2.1 to produce extreme precipitation weather fields. Figures 14 and 15 show the generation of extreme precipitation weather fields: the first-left column of images are the target weather fields produced by the weather generator based in some criteria, in this experiment, we arbitrarily selected those figures. Each row of Figures 14 and 15 contain the results of conditioning the Direct Sampling on the three precipitation level maps — 100, 250, and 500 -year return precipitation levels — depicted in Figure 13. The yellow areas in the second, fourth, and sixth columns of images depict the ROIs: a rectangle, two rectangles, and a more complex shape estimated by selecting the areas from the original target image with more than 90th percentile of precipitation values. Each ROI contains a set of control points, in practice, users can define arbitrary locations for those control points within the ROI. In this experiment, the control points were randomly located. Observe that the value that each control point could take is defined by the random process f (Section 3.2.1). For instance, Figure 14 shows the case where the control point values are estimated by defining a random variable f_i uniformly distributed with limits given by the maximum value presented in the training image, and the return precipitation level map value for the location i of the control point. On the other hand, Figure 15 shows the case where we define f_i to be the precipitation value at the same location within the return level precipitation map, i.e., $f_i = \mathbf{M}_i$. The third, fifth, and seventh columns from Figures 14 and 15 show the generated precipitation weather fields with extreme precipitation values correlated with the information provided by the return precipitation level map within the ROI. The color bars on the right side of the plot inform the precipitation values in mm/day, where its maximum value is given by the 95th percentile of the whole return precipitation level map in analysis.

Notice that the statistical and structural properties of the resulting simulation in the region outside the ROI will be correlated with those in the training image because

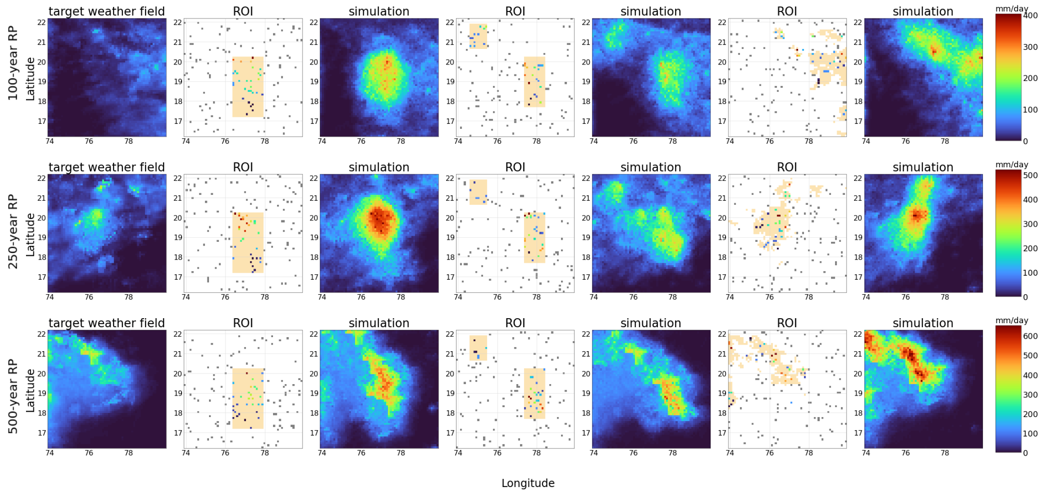

Figure 15. Extreme precipitation generation using the Direct Sampling and control points located in the yellow area of the ROI, conditioned on a 100, 250 and 500-year return precipitation level maps. Each control point value is the value at the control point location in the return precipitation level map.

of the use of the conditioning points. To analyze how the extreme generation alters the statistical and connectivity metrics of the original training image, we show in Figure 16 the results of comparing the statistical and connectivity properties from one hundred simulations of extreme precipitation weather field generation for the case where each control point value is the value at the control point location in the return precipitation level map, vs. the statistical and connectivity properties from the original training image. In this case, we used as a training image a precipitation weather field that contains a maximum precipitation value of 409.9 mm/day (depicted in Figure 16), and an ROI defined by the rectangle shown in the second column of Figure 15. Also, we used the return precipitation level map corresponding to 500-year RP shown in Figure 13. This return precipitation level map contains, within the ROI, maximum and average precipitation values of $1029, 5$ mm/day and 340 mm/day respectively.

The first row of Figure 16 shows the mean and standard deviation images of one hundred simulations. It is possible to observe that — on average — Direct Sampling generates higher values within the ROI, and it generates lower values outside the ROI, also the variability of values is lower outside the ROI than within the ROI. The QQ-plot and the eCDF metrics show how the distribution of precipitation values is shifted upwards to match the maximum precipitation value within the ROI location in the return precipitation level map ($1029, 5$ mm/day). The variogram reflects the increase in variability as a function of a lag, due to generated extreme precipitation values. The connectivity metrics C_2 and S_2 show an increase in the connectivity metric values for high precipitation values, i.e., the probability of having a path between two random points within the area of high precipitation is bigger, and there is a decrease in the probability of connectivity of low precipitation values, which makes sense because the algorithm is increasing the precipitation values within the ROI. The connectivity metrics also show that the connectivity properties of the middle precipitation values are well preserved by the simulations.

To provide a contrast between the proposed extreme weather generation approach and a more traditional method such as quantile mappings, we also employed the latter

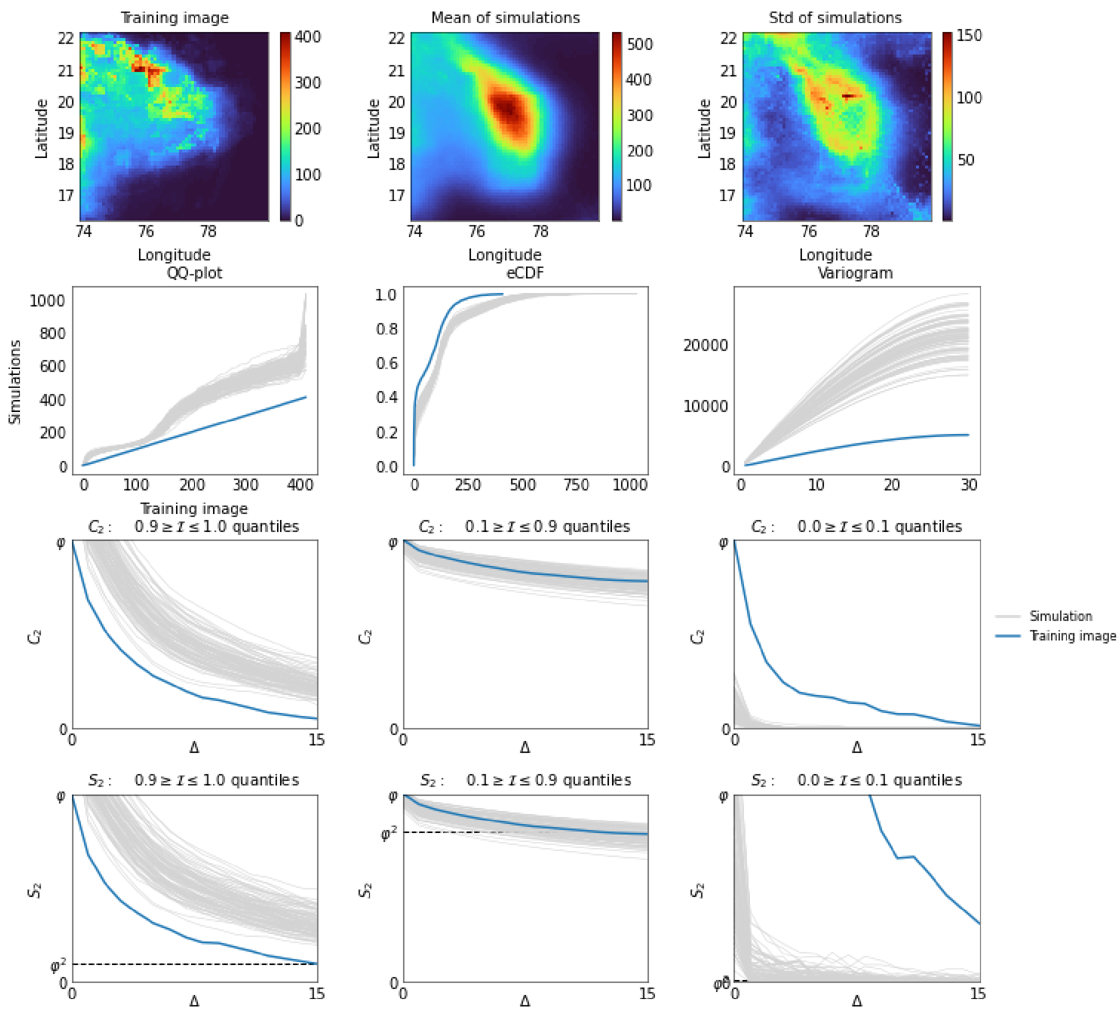

Figure 16. Analysis of the extreme precipitation fields generated with Direct Sampling using statistical and connectivity metrics.

on the results of Direct Sampling simulations to generate extremes. This is described in the Supplemental Material for completeness.

5 Conclusions

Weather generators based on resampling are powerful tools for generating new time series of weather data such that the simulated weather data has similar statistics to the original one. Those weather generators are easy to implement, do not rely on parametric distributions, and are fully data-driven. They will always generate new time series by assembling copies of weather fields found in the original dataset in such a way that it is possible to reproduce the monthly, seasonal, or annual statistics found in the observations. They perfectly reproduce the interplay dynamics among weather variables in the spatial domain, because of the resampling process. However, the spatial variability is constrained to the choices made by its resampling strategy. Furthermore, the out-of-sample extreme weather field generation is no longer possible because the resampling is limited to the observations by definition. In this work, we show how those issues can be addressed by including the Direct Sampling algorithm as part of those weather generators. Direct Sampling is an algorithmic approach based on pattern matching for producing simulations from a training image with similar statistical properties. Such an algorithm is conceptually simple to understand and implement, and it can be conditioned on a set of conditioning points in such a way that the simulations respect the information provided by those points. We propose to improve the spatial variability of the simulations provided by weather generators based on resampling by post-processing with the Direct Sampling algorithm, each simulated weather field produced by the weather generator. To keep the coherence among the interplay dynamics of weather fields, we share the locations of the random conditioning points among all the weather fields at each timestep. We also propose to use Direct Sampling jointly with return period analysis for generating out-of-sample extreme weather fields in specific locations. We conducted a series of experiments using precipitation, temperature and cloud-cover weather data to demonstrate the spatial variability enhancement and precipitation data for generating extreme precipitation events in a region in north-west India . Our experimental results show that the presented approach can be useful in practice for improving the spatial variability and out-of-sample weather field generation of weather generators based on resampling. Future work will include extreme generation in a multivariate setting, the use of physics as a validation metric, and the analysis of the relationship between the sampling frequency of control points within the training image and the spatial distribution of extremes.

Acknowledgments

We would like to express our sincere gratitude to the anonymous reviewers for their invaluable contributions to our paper. Their thorough evaluation, constructive feedback, and thoughtful suggestions have greatly improved the quality and the final version of manuscript. We are immensely grateful for their time, effort, and valuable input.

Appendix A Open Research

The weather data used for the experiments described in the study are 1) the IMERG dataset (Huffman et al., 2019) available at NASA Earthdata via `https://doi.org/10.5067/GPM/IMERGDF/DAY/06` This dataset is intended for public access and use. No license information was provided (All NASA-produced data from the GPM mission is made freely available for the public to use) ; and 2) the ERA5 (Hersbach et al., 2018) data set available at Copernicus via `https://doi.org/10.24381/cds.adbb2d47` with license available at `https://apps.ecmwf.int/datasets/licences/copernicus/`. We used Python version 3.9.13 (van Rossum, 1995) with license available at `https://docs.python.org/3/license.html` for a Python implementation of the weather generator `https://github.com/IBM/IBMWeatherGen/`, and for a Python implementation of the Direct Sampling routine at `https://wp.unil.ch/gaia/mps/ds-matlab/`. There is also an open-source implementation of the Direct Sampling algorithm called QuickSampling (Gravey & Mariethoz, 2020), which is known for its fast performance. We used Matplotlib 3.5.1 with license at `https://matplotlib.org/stable/users/project/license.html` and Seaborn 0.11.2 with license at `https://github.com/mwaskom/seaborn/blob/master/LICENSE.md` for creating the Figures. Statistical analysis were carried out with Statsmodels 0.13.2 with license at `https://www.statsmodels.org/` and connectivity metric analysis with GooseEYE `https://gooseeye.readthedocs.io/` with license available at `https://github.com/tdegeus/GooseEYE/blob/main/LICENSE`.

References

Ailliot, P., Allard, D., Monbet, V., & Naveau, P. (2015). Stochastic weather generators: an overview of weather type models. *Journal de la Société Française de*

Statistique, *156*(1), 101–113.
Allard, D., & Bourotte, M. (2015). Disaggregating daily precipitations into hourly values with a transformed censored latent gaussian process. *Stochastic environmental research and risk assessment*, *29*(2), 453–462.
Apipattanavis, S., Podestá, G., Rajagopalan, B., & Katz, R. W. (2007). A semiparametric multivariate and multisite weather generator. *Water Resources Research*, *43*(11).
Benoit, L., & Mariethoz, G. (2017). Generating synthetic rainfall with geostatistical simulations. *Wiley Interdisciplinary Reviews: Water*, *4*(2), e1199.
Brisson, N., Launay, M., Mary, B., & Beaudoin, N. (2009). Conceptual basis, formalisations and parameterization of the stics crop model: Editions quae. *Collection Update Sciences and technologies*.
Brunner, M. I., Seibert, J., & Favre, A.-C. (2016). Bivariate return periods and their importance for flood peak and volume estimation. *Wiley Interdisciplinary Reviews: Water*, *3*(6), 819–833.
Buishand, T. A., & Brandsma, T. (2001). Multisite simulation of daily precipitation and temperature in the rhine basin by nearest-neighbor resampling. *Water Resources Research*, *37*(11), 2761–2776.
Carey-Smith, T., Sansom, J., & Thomson, P. (2014). A hidden seasonal switching model for multisite daily rainfall. *Water Resources Research*, *50*(1), 257–272.
Chen, J., Brissette, F., & Leconte, R. (2012). Weagets–a matlab-based daily scale weather generator for generating precipitation and temperature. *Procedia Environmental Sciences*, *13*, 2222–2235.
Cui, Z., Chen, Q., Liu, G., Mariethoz, G., & Ma, X. (2021). Hybrid parallel framework for multiple-point geostatistics on tianhe-2: A robust solution for large-scale simulation. *Computers & Geosciences*, *157*, 104923.
De Haan, L., & Ferreira, A. (2007). *Extreme value theory: an introduction.* Springer Science & Business Media.
Exum, N. G., Betanzo, E., Schwab, K. J., Chen, T. Y., Guikema, S., & Harvey, D. E. (2018). Extreme precipitation, public health emergencies, and safe drinking water in the usa. *Current environmental health reports*, *5*, 305–315.
Gravey, M., & Mariethoz, G. (2020). Quicksampling v1. 0: a robust and simplified pixel-based multiple-point simulation approach. *Geoscientific Model Development*, *13*(6), 2611–2630.
Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., ... Dee, J.-N., D ans Thépaut (2018). Era5 hourly data on single levels from 1959 to present. copernicus climate change service (c3s) climate data store (cds) [dataset]. *10.24381/cds.adbb2d47*.
Huang, T., Li, X., Zhang, T., & Lu, D.-T. (2013). Gpu-accelerated direct sampling method for multiple-point statistical simulation. *Computers & Geosciences*, *57*, 13–23.
Huffman, G., Stocker, E., Bolvin, D., Nelkin, E., & Tan, J. (2019). Gpm imerg final precipitation l3 1 day 0.1 degree x 0.1 degree v06, edited by andrey savtchenko, greenbelt, md, goddard earth sciences data and information services center (ges disc) [dataset]. *doi. org/10.5067/GPM/IMERG DF/DAY/06*.
Jha, S. K., Mariethoz, G., Evans, J., McCabe, M. F., & Sharma, A. (2015). A space and time scale-dependent nonlinear geostatistical approach for downscaling daily precipitation and temperature. *Water Resources Research*, *51*(8), 6244–6261.
Jha, S. K., Mariethoz, G., Evans, J. P., & McCabe, M. F. (2013). Demonstration of a geostatistical approach to physically consistent downscaling of climate modeling simulations. *Water Resources Research*, *49*(1), 245–259.
Kenabatho, P. K., McIntyre, N., Chandler, R., & Wheater, H. (2012). Stochastic simulation of rainfall in the semi-arid limpopo basin, botswana. *International*

Journal of Climatology, *32*(7), 1113–1127.
Kilsby, C. G., Jones, P., Burton, A., Ford, A., Fowler, H. J., Harpham, C., ... Wilby, R. (2007). A daily weather generator for use in climate change studies. *Environmental Modelling & Software*, *22*(12), 1705–1719.
Kim, Y., Katz, R., Rajagopalan, B., Podestá, G., & Furrer, E. (2012). Reducing overdispersion in stochastic weather generators using a generalized linear modeling approach. *Climate research*, *53*(1), 13–24.
Kolokotroni, M., Ren, X., Davies, M., & Mavrogianni, A. (2012). London's urban heat island: Impact on current and future energy consumption in office buildings. *Energy and buildings*, *47*, 302–311.
Kwon, H.-H., Lall, U., & Khalil, A. F. (2007). Stochastic simulation model for nonstationary time series using an autoregressive wavelet decomposition: Applications to rainfall and temperature. *Water Resources Research*, *43*(5).
Lall, U., & Sharma, A. (1996). A nearest neighbor bootstrap for time series resampling. *Water Resour. Res*, *32*(3), 679–693.
Lee, D., An, H., Lee, Y., Lee, J., Lee, H.-S., & Oh, H.-S. (2010). Improved multisite stochastic weather generation with applications to historical data in south korea. *Asia-Pacific Journal of Atmospheric Sciences*, *46*(4), 497–504.
Lennartsson, J., Baxevani, A., & Chen, D. (2008). Modelling precipitation in sweden using multiple step markov chains and a composite model. *Journal of hydrology*, *363*(1-4), 42–59.
Mariethoz, G., & Caers, J. (2014). *Multiple-point geostatistics: stochastic modeling with training images.* John Wiley & Sons.
Mariethoz, G., Renard, P., & Straubhaar, J. (2010). The direct sampling method to perform multiple-point geostatistical simulations. *Water Resources Research*, *46*(11).
Meerschman, E., Pirot, G., Mariethoz, G., Straubhaar, J., Van Meirvenne, M., & Renard, P. (2013). A practical guide to performing multiple-point statistical simulations with the direct sampling algorithm. *Computers & Geosciences*, *52*, 307–324.
Oliveira, D. H. M. C., Lima, K. C., & Spyrides, M. H. C. (2021). Rainfall and streamflow extreme events in the são francisco hydrographic region. *International Journal of Climatology*, *41*(2), 1279–1291.
Opitz, T., Allard, D., & Mariethoz, G. (2021). Semi-parametric resampling with extremes. *Spatial Statistics*, *42*, 100445.
Oriani, F., Mehrotra, R., Mariethoz, G., Straubhaar, J., Sharma, A., & Renard, P. (2018). Simulating rainfall time-series: how to account for statistical variability at multiple scales? *Stochastic environmental research and risk assessment*, *32*(2), 321–340.
Oriani, F., Ohana-Levi, N., Marra, F., Straubhaar, J., Mariethoz, G., Renard, P., ... Morin, E. (2017). Simulating small-scale rainfall fields conditioned by weather state and elevation: A data-driven approach based on rainfall radar images. *Water Resources Research*, *53*(10), 8512–8532.
Oriani, F., Straubhaar, J., Renard, P., & Mariethoz, G. (2014). Simulation of rainfall time series from different climatic regions using the direct sampling technique. *Hydrology and Earth System Sciences*, *18*(8), 3015–3031.
Peleg, N., Fatichi, S., Paschalis, A., Molnar, P., & Burlando, P. (2017). An advanced stochastic weather generator for simulating 2-d high-resolution climate variables. *Journal of Advances in Modeling Earth Systems*, *9*(3), 1595–1627.
Peleg, N., Molnar, P., Burlando, P., & Fatichi, S. (2019). Exploring stochastic climate uncertainty in space and time using a gridded hourly weather generator. *Journal of Hydrology*, *571*, 627–641.
Rajagopalan, B., & Lall, U. (1999). A k-nearest-neighbor simulator for daily precipitation and other weather variables. *Water resources research*, *35*(10), 3089–3101.

Rajagopalan, B., Lall, U., Tarboton, D. G., & Bowles, D. (1997). Multivariate nonparametric resampling scheme for generation of daily weather variables. *Stochastic Hydrology and Hydraulics*, *11*(1), 65–93.

Renard, P., & Allard, D. (2013). Connectivity metrics for subsurface flow and transport. *Advances in Water Resources*, *51*, 168–196.

Revi, A. (2008). Climate change risk: an adaptation and mitigation agenda for indian cities. *Environment and urbanization*, *20*(1), 207–229.

Richardson, C. W. (1981). Stochastic simulation of daily precipitation, temperature, and solar radiation. *Water resources research*, *17*(1), 182–190.

Samuels, R., Rimmer, A., & Alpert, P. (2009). Effect of extreme rainfall events on the water resources of the jordan river. *Journal of Hydrology*, *375*(3-4), 513–523.

Semenov, M. A., Brooks, R. J., Barrow, E. M., & Richardson, C. W. (1998). Comparison of the wgen and lars-wg stochastic weather generators for diverse climates. *Climate research*, *10*(2), 95–107.

Steinschneider, S., & Brown, C. (2013). A semiparametric multivariate, multisite weather generator with low-frequency variability for use in climate risk assessments. *Water resources research*, *49*(11), 7205–7220.

Steinschneider, S., Ray, P., Rahat, S. H., & Kucharski, J. (2019). A weather-regime-based stochastic weather generator for climate vulnerability assessments of water systems in the western united states. *Water Resources Research*, *55*(8), 6923–6945.

Todorovic, P., & Woolhiser, D. A. (1975). A stochastic model of n-day precipitation. *Journal of Applied Meteorology (1962-1982)*, 17–24.

Torquato, S., Beasley, J., & Chiew, Y. (1988). Two-point cluster function for continuum percolation. *The Journal of chemical physics*, *88*(10), 6540–6547.

Torquato, S., & Haslach Jr, H. (2002). Random heterogeneous materials: microstructure and macroscopic properties. *Appl. Mech. Rev.*, *55*(4), B62–B63.

Trigo, R. M., & Palutikof, J. P. (1999). Simulation of daily temperatures for climate change scenarios over portugal: a neural network model approach. *Climate Research*, *13*(1), 45–59.

van Rossum, G. (1995, May). *Python tutorial* (software No. CS-R9526). Amsterdam: Centrum voor Wiskunde en Informatica (CWI).

Vogel, R. M., & Castellarin, A. (2017). Risk, reliability, and return periods and hydrologic design. *Handbook of Applied Hydrology; Singh, VP, Ed.; McGraw-Hill Book Company: New York, NY, USA*.

Wang, P., Deng, X., Zhou, H., & Qi, W. (2018). Responses of urban ecosystem health to precipitation extreme: A case study in beijing and tianjin. *Journal of Cleaner Production*, *177*, 124–133.

Wilcox, C., Aly, C., Vischel, T., Panthou, G., Blanchet, J., Quantin, G., & Lebel, T. (2021). Stochastorm: A stochastic rainfall simulator for convective storms. *Journal of Hydrometeorology*, *22*(2), 387–404.

Wilks, D. (1998). Multisite generalization of a daily stochastic precipitation generation model. *journal of Hydrology*, *210*(1-4), 178–191.

Wilks, D. S., & Wilby, R. L. (1999). The weather generation game: a review of stochastic weather models. *Progress in physical geography*, *23*(3), 329–357.

Wojcik, R., McLaughlin, D., Konings, A. G., & Entekhabi, D. (2009). Conditioning stochastic rainfall replicates on remote sensing data. *IEEE transactions on geoscience and remote sensing*, *47*(8), 2436–2449.

Woldemichael, A. T., Hossain, F., Pielke Sr, R., & Beltrán-Przekurat, A. (2012). Understanding the impact of dam-triggered land use/land cover change on the modification of extreme precipitation. *Water Resources Research*, *48*(9).

Zhou, Q., Panduro, T. E., Thorsen, B. J., & Arnbjerg-Nielsen, K. (2013). Adaption to extreme rainfall with open urban drainage system: An integrated hydrological cost-benefit analysis. *Environmental management*, *51*, 586–601.