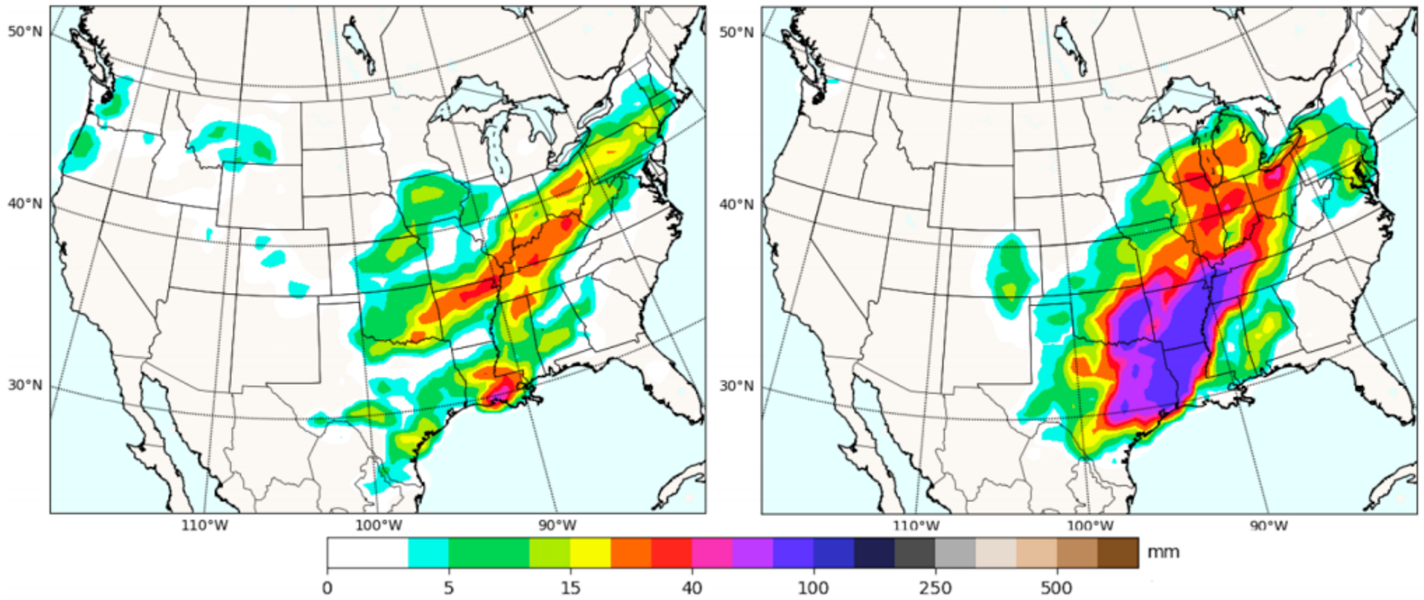
ExGAN:
Adversarial Generation of Extreme Samples

Siddharth Bhatia*, Arjit Jain*, Bryan Hooi

* Equal Contribution



Left: Existing GAN-based approaches fit bulk of the data
Generate typical data samples
Shown by rainfall patterns which have low to moderate rainfall

Right: Our approach tries to fit the extreme tail of the distribution
Generates extreme data samples with varying severity
Shown by extreme rainfall with spatial patterns resembling real floods

Motivation

To model extreme events in order to evaluate and mitigate their risk

Applications in extreme weather events, financial crashes, and managing unexpectedly high demand for online services

To be able to generate a wide range of extreme scenarios

Can be used by domain experts to understand the nature of extreme events

Can be used to perform stress-testing to ensure the system remains stable under a wide range of extreme but realistic scenarios

Problem Statement

How can we generate a wide range of extreme but realistic scenarios?

What does it mean to be extreme?

How do we *measure* extremeness?

Examples: Database Management Systems

End Goal: Resilience against high query loads

Extremeness Measure: Number of queries per second

Want to generate: Rapidly arriving query loads with realistic access patterns

Examples: Rainfall Analysis

End Goal: Information about severe floods

Extremeness Measure: Total rainfall

Want to generate: High severity floods with realistic rainfall patterns

Extremeness Probability

In hydrology, a 100-year flood is defined as a flood that has a 1 in 100 chance of being exceeded in any given year

In a similar way, for conditional generation, we define extremeness probability τ which represents how extreme the user wants their sampled data to be.

For example, $\tau = 0.01$ represents generating an event whose extremeness measure is only exceeded 1% of the time

Problem Statement, formally

We are given:

A training dataset $\mathbf{x}_1, \cdots, \mathbf{x}_n \sim \mathcal{D}$, a user defined extremeness $\mathsf{E}(\mathbf{x})$ measure, and a user specified extremeness $\tau \in (0, 1)$ probability

We want to generate samples $\mathbf{x}'$ that are:

1. Realistic, i.e. hard to distinguish from the training data
2. Extreme at the given level, i.e. $P_{\mathbf{x}\sim\mathcal{D}}(\mathsf{E}(\mathbf{x}) > \mathsf{E}(\mathbf{x}'))$ is as close as possible to τ

Challenges

1. **Lack of training examples:** In a moderately sized dataset, the rarity of "extreme" samples means that it is typically infeasible to train a generative model only on these extreme samples

2. **Conditional Generation:** We need to generate extreme samples at any given, user-specified extremeness probability

Our Approach

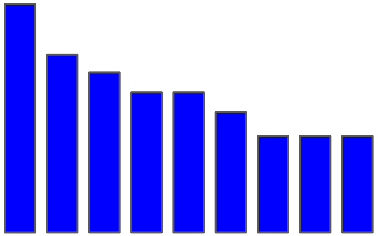
1. **Distribution Shifting**

   Gradually shift the data distribution in the direction of increasing extremeness.
   Allows us to fit a GAN in a robust and stable manner, while fitting the tail of the distribution, rather than its bulk

1. **Extreme Value Theory (EVT) based Conditional Generation**

   Train a conditional GAN, conditioned on the extremeness measure
   Use EVT analysis, along with keeping track of the amount of distribution shifting performed, to generate new samples at the given extremeness probability

Distribution Shifting

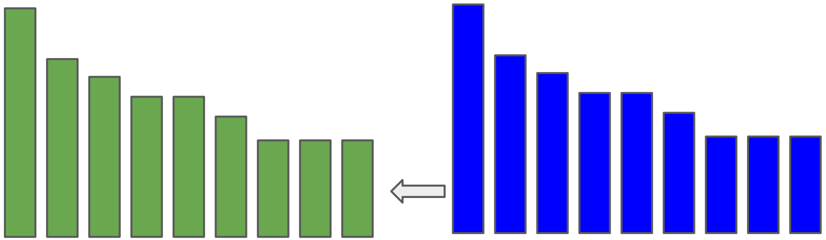


Algorithm 1: Distribution Shifting

Input: dataset $\mathcal{X}$, extremeness measure E, shift parameter c, iteration count k

Sort $\mathcal{X}$ in decreasing order of extremeness

Distribution Shifting

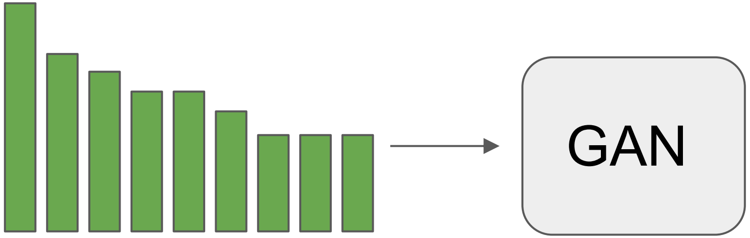


Algorithm 1: Distribution Shifting

---

1 **Input**: dataset $\mathcal{X}$, extremeness measure E, shift parameter c, iteration count k
2 Sort $\mathcal{X}$ in decreasing order of extremeness
3 Initialize $\mathcal{X}_s \leftarrow \mathcal{X}$

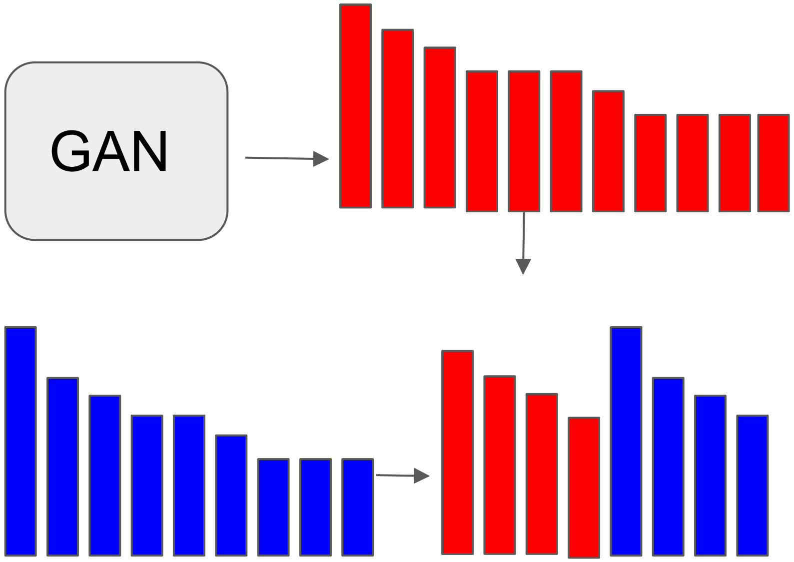
Distribution Shifting



Algorithm 1: Distribution Shifting

Input: dataset $\mathcal{X}$, extremeness measure E, shift parameter c, iteration count k
Sort $\mathcal{X}$ in decreasing order of extremeness
Initialize $\mathcal{X}_s \leftarrow \mathcal{X}$
for $i \leftarrow 1$ *to* k **do**
    $\triangleright$ **Shift the data distribution by a factor of c:**
    Train DCGAN G and D on $\mathcal{X}_s$

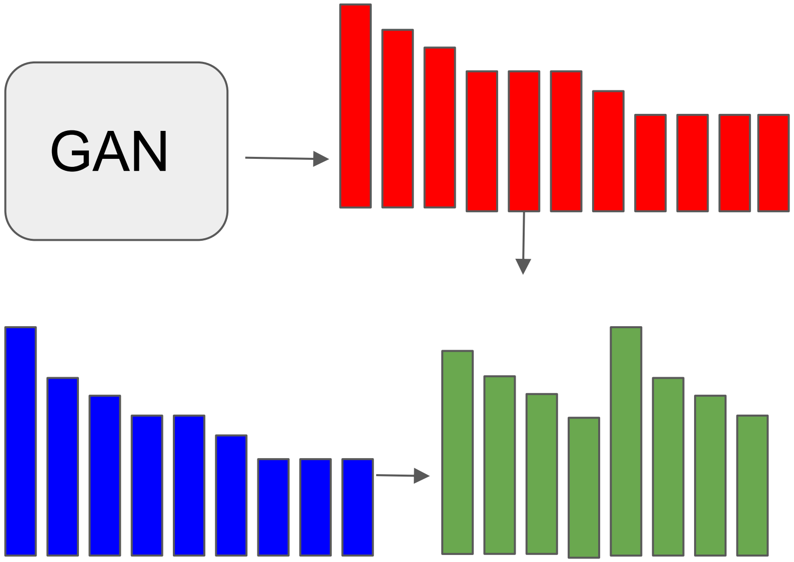
Distribution Shifting



Algorithm 1: Distribution Shifting

Input: dataset $\mathcal{X}$, extremeness measure E, shift parameter c, iteration count k
Sort $\mathcal{X}$ in decreasing order of extremeness
Initialize $\mathcal{X}_s \leftarrow \mathcal{X}$
for $i \leftarrow 1$ *to* k **do**
    ▷ **Shift the data distribution by a factor of c:**
    Train DCGAN G and D on $\mathcal{X}_s$
    $\mathcal{X}_s \leftarrow$ top $\lfloor c^i \cdot n \rfloor$ extreme samples of $\mathcal{X}$
    Generate $\lceil (n - \lfloor c^i \cdot n \rfloor) \cdot \frac{1}{c} \rceil$ data points using G, and insert most extreme $n - \lfloor c^i \cdot n \rfloor$ samples into $\mathcal{X}_s$

Distribution Shifting



Algorithm 1: Distribution Shifting

Input: dataset $\mathcal{X}$, extremeness measure E, shift parameter c, iteration count k
Sort $\mathcal{X}$ in decreasing order of extremeness
Initialize $\mathcal{X}_s \leftarrow \mathcal{X}$
for $i \leftarrow 1$ *to* k **do**
  ▷ **Shift the data distribution by a factor of c:**
  Train DCGAN G and D on $\mathcal{X}_s$
  $\mathcal{X}_s \leftarrow$ top $\lfloor c^i \cdot n \rfloor$ extreme samples of $\mathcal{X}$
  Generate $\lceil (n - \lfloor c^i \cdot n \rfloor) \cdot \frac{1}{c} \rceil$ data points using G, and insert most extreme $n - \lfloor c^i \cdot n \rfloor$ samples into $\mathcal{X}_s$

Extreme Value Theory

Generalized Pareto Distribution (GPD)

The parameters of GPD are its scale σ and its shape ξ. The cumulative distribution function (CDF) of the GPD is:

$$G_{\sigma,\xi}(x) = \begin{cases} 1 - (1 + \frac{\xi \cdot x}{\sigma})^{-1/\xi} & \text{if } \xi \neq 0 \\ 1 - \exp(-\frac{x}{\sigma}) & \text{if } \xi = 0 \end{cases}$$

Extreme Value Theory (EVT)

Peaks over Threshold

A theorem in EVT states that the excess over a sufficiently large threshold u, denoted by $X - u$, is likely to follow a Generalized Pareto Distribution (GPD) with parameters $\sigma(u), \xi$

In practice, the threshold u is set a value around the 95th percentile.

EVT based Conditional Generation

After k shifts, the adjusted extremeness probability becomes $\tau' = \tau/c^k$

Algorithm 2: EVT-based Conditional Generation

1 **Input**: shifted dataset $\mathcal{X}_s$, extremeness measure E, adjusted extremeness probability τ'

EVT based Conditional Generation

We then use the Peaks over Threshold method, and estimate GPD parameters

Algorithm 2: EVT-based Conditional Generation

1 **Input**: shifted dataset $\mathcal{X}_s$, extremeness measure E, adjusted extremeness probability τ'

2 Compute extremeness values $e_i = \mathsf{E}(\mathbf{x_i}) \, \forall \, \mathbf{x_i} \in \mathcal{X}_s$

3 Fit GPD parameters σ, ξ using maximum likelihood (Grimshaw 1993) on $e_1, \cdots, e_n$

EVT based Conditional Generation

In addition to the data samples, D_s takes in a second input which is e for a generated sample $G_s(\mathbf{z}, e)$ and $\mathsf{E}(\mathbf{x})$ for a real sample x

Algorithm 2: EVT-based Conditional Generation

1 **Input**: shifted dataset $\mathcal{X}_s$, extremeness measure E, adjusted extremeness probability τ'

2 Compute extremeness values $e_i = \mathsf{E}(\mathbf{x_i}) \; \forall \; \mathbf{x_i} \in \mathcal{X}_s$

3 Fit GPD parameters σ, ξ using maximum likelihood (Grimshaw 1993) on $e_1, \cdots, e_n$

4 Train conditional DCGAN (G_s and D_s) on $\mathcal{X}_s$ where the conditioning input for G_s is sampled from a GPD with parameters σ, ξ

EVT based Conditional Generation

An additional loss is added to the GAN objective:

$$\mathcal{L}_{\text{ext}} = \mathbb{E}_{\mathbf{z},e}\left[\frac{|e - \mathsf{E}(G_s(\mathbf{z},e))|}{e}\right]$$

where z is sampled from multivariate standard normal distribution and e is sampled from a GPD with parameters σ , ξ

Algorithm 2: EVT-based Conditional Generation

1. **Input**: shifted dataset $\mathcal{X}_s$, extremeness measure E, adjusted extremeness probability τ'
2. Compute extremeness values $e_i = \mathsf{E}(\mathbf{x_i})\ \forall\ \mathbf{x_i} \in \mathcal{X}_s$
3. Fit GPD parameters σ, ξ using maximum likelihood (Grimshaw 1993) on $e_1, \cdots, e_n$
4. Train conditional DCGAN (G_s and D_s) on $\mathcal{X}_s$ where the conditioning input for G_s is sampled from a GPD with parameters σ, ξ

EVT based Conditional Generation

Using the inverse CDF of the GPD, we determine the extremeness level e' that corresponds to an extremeness probability of τ'

Algorithm 2: EVT-based Conditional Generation

---

1. **Input**: shifted dataset $\mathcal{X}_s$, extremeness measure E, adjusted extremeness probability τ'
2. Compute extremeness values $e_i = \mathsf{E}(\mathbf{x_i}) \; \forall \; \mathbf{x_i} \in \mathcal{X}_s$
3. Fit GPD parameters σ, ξ using maximum likelihood (Grimshaw 1993) on $e_1, \cdots, e_n$
4. Train conditional DCGAN (G_s and D_s) on $\mathcal{X}_s$ where the conditioning input for G_s is sampled from a GPD with parameters σ, ξ
5. Extract required extremeness level: $e' \leftarrow G_{\sigma,\xi}^{-1}(1 - \tau')$
6. Sample from G_s conditioned on extremeness level e'

Baseline

The baseline is a DCGAN trained over all the images in the dataset, combined with rejection sampling

Use EVT as in our framework to compute the extremeness level $e = G^{-1}_{\sigma,\xi}(1-\tau)$ that corresponds to an extremeness probability of τ

Repeatedly generate images until one is found that satisfies the extremeness criterion within 10% error; that is, we reject the image x if

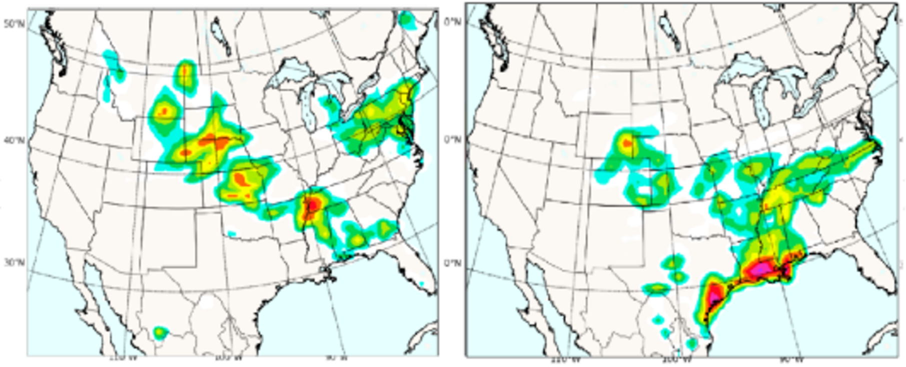
$$\left|\frac{e - \mathsf{E}(\mathbf{x})}{e}\right| > 0.1$$

Dataset

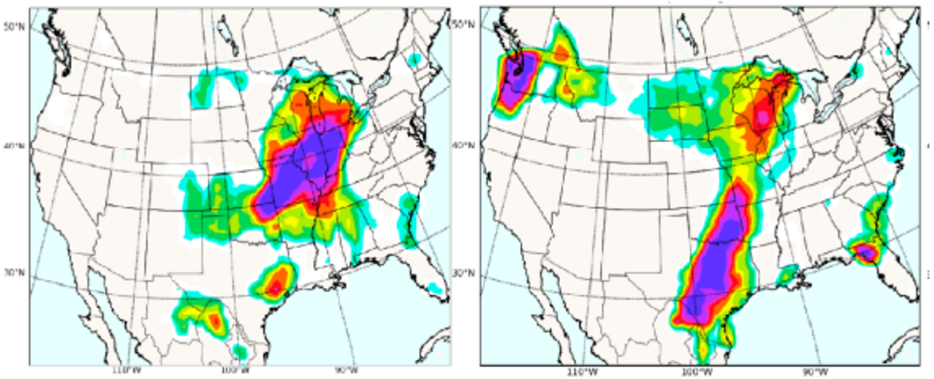
Daily US Precipitation Data: Records the amount of rainfall over a spatial grid

For Training: Data for the duration January 2010 - December 2016

For Testing: Extreme Data $(\tau < 0.05)$ in the duration January 2017 - August 2020



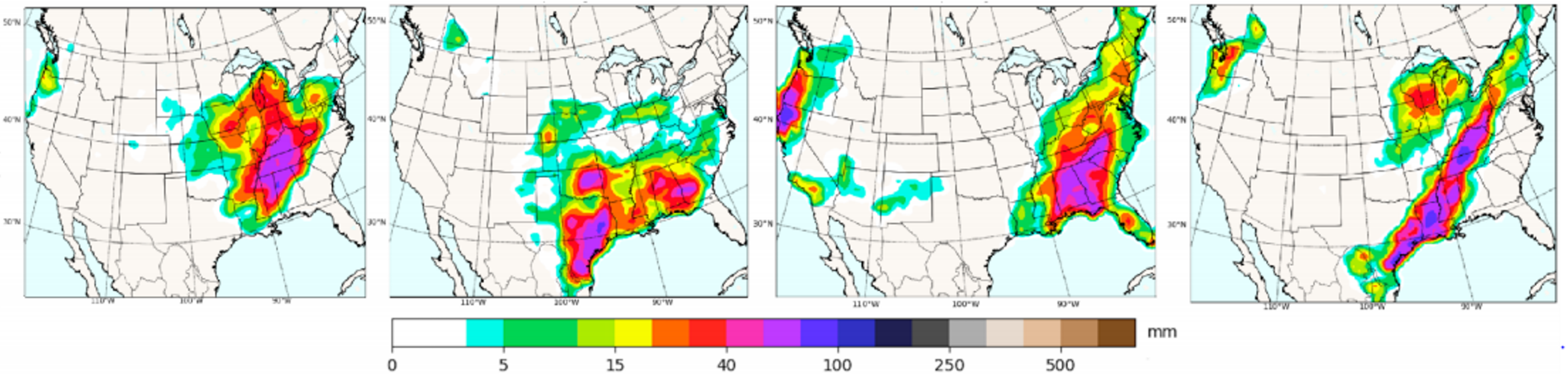
Normal examples from the dataset



Extreme examples from the dataset

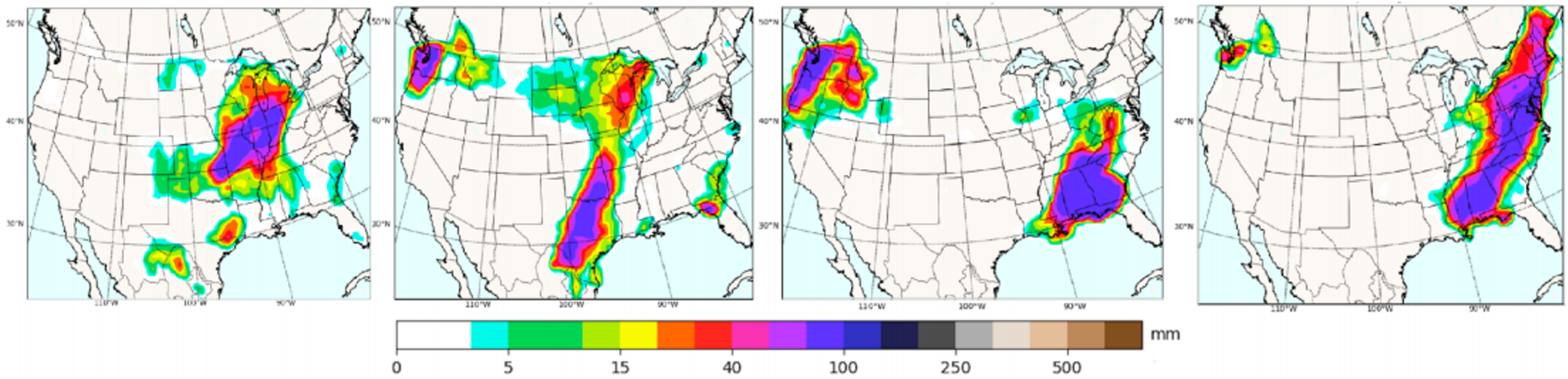
ExGAN generated Samples

At $\tau = 0.001$



ExGAN generated Samples

At $\tau = 0.0001$



Evaluation Metrics

Fréchet Inception Distance (FID)

We train an autoencoder on the test data. Use the statistics on its bottleneck activations, on the generated and real samples, to compute FID

Reconstruction Loss

We try to reconstruct the test data by optimizing over the latent space vector

Method	FID	Reconstruction Loss
DCGAN	0.0406 ± 0.0063	0.0292
ExGAN	0.0236 ± 0.0037	0.0172

(Lower is better)

Sampling Time

We report the time taken to generate 100 samples for different extremeness probabilities

DCGAN could not generate even one sample for extremeness probabilities $\tau = 0.001$ and $\tau = 0.0001$ in 1 hour

ExGAN is scalable and generates extreme samples in constant time

Method	**Extremeness Probability (τ)**			
	0.05	0.01	0.001	0.0001
DCGAN	$1.230s$	$7.564s$	$-$	$-$
ExGAN	$0.002s$	$0.002s$	$0.002s$	$0.002s$

Conclusion

- We proposed a novel deep learning-based approach for generating extreme data using distribution-shifting and EVT analysis
- We demonstrated how our approach is scalable and able to generate extreme samples in constant time
- Our experimental results show that ExGAN generates realistic samples based on both visual inspection and quantitative metrics

[2009.08454] ExGAN: Adversarial Generation of Extreme Samples (arxiv.org)

Stream-AD/ExGAN: Adversarial Generation of Extreme Samples (github.com)