# ExGAN: Adversarial Generation of Extreme Samples

**Siddharth Bhatia[1]\*, Arjit Jain[2]\*, Bryan Hooi[1]**

[1]National University of Singapore
[2]IIT Bombay
siddharth@comp.nus.edu.sg, arjit@cse.iitb.ac.in, bhooi@comp.nus.edu.sg

## Abstract

Mitigating the risk arising from extreme events is a fundamental goal with many applications, such as the modelling of natural disasters, financial crashes, epidemics, and many others. To manage this risk, a vital step is to be able to understand or generate a wide range of extreme scenarios. Existing approaches based on Generative Adversarial Networks (GANs) excel at generating realistic samples, but seek to generate *typical* samples, rather than extreme samples. Hence, in this work, we propose ExGAN, a GAN-based approach to generate realistic and extreme samples. To model the extremes of the training distribution in a principled way, our work draws from Extreme Value Theory (EVT), a probabilistic approach for modelling the extreme tails of distributions. For practical utility, our framework allows the user to specify both the desired extremeness measure, as well as the desired extremeness probability they wish to sample at. Experiments on real US precipitation data show that our method generates realistic samples, based on visual inspection and quantitative measures, in an efficient manner. Moreover, generating increasingly extreme examples using ExGAN can be done in constant time (with respect to the extremeness probability $\tau$), as opposed to the $\mathcal{O}(\frac{1}{\tau})$ time required by the baseline approach.

## 1 Introduction

Modelling extreme events in order to evaluate and mitigate their risk is a fundamental goal with a wide range of applications, such as extreme weather events, financial crashes, and managing unexpectedly high demand for online services. A vital part of mitigating this risk is to be able to understand or generate a wide range of extreme scenarios. For example, in many applications, stress-testing is an important tool, which typically requires testing a system on a wide range of extreme but realistic scenarios, to ensure that the system can successfully cope with such scenarios. This leads to the question: how can we generate a wide range of extreme but realistic scenarios, for the purpose of understanding or mitigating their risk?

Recently, Generative Adversarial Networks (GANs) and their variants have led to tremendous interest, due to their ability to generate highly realistic samples. On the other

hand, existing GAN-based methods generate *typical* samples, i.e. samples that are similar to those drawn from the bulk of the distribution. Our work seeks to address the question: how can we design deep learning-based models which can generate samples that are not just realistic, but also extreme (with respect to any user-specified measure)? Answering this question would allow us to generate extreme samples that can be used by domain experts to assist in their understanding of the nature of extreme events in a given application. Moreover, such extreme samples can be used to perform stress-testing of existing systems, to ensure that the systems remain stable under a wide range of extreme but realistic scenarios.

Our work relates to the recent surge of interest in making deep learning algorithms reliable even for safety-critical applications such as medical applications, self-driving cars, aircraft control, and many others. Toward this goal, our work explores how deep generative models can be used for understanding and generating the extremes of a distribution, for any user-specified extremeness probability, rather than just generating typical samples as existing GAN-based approaches do.

More formally, our problem is as follows: Given a data distribution and a criterion to measure extremeness of any sample in this data, can we generate a diverse set of realistic samples with any given extremeness probability? Consider a database management setting with queries arriving over time; users are typically interested in resilience against high query loads, so they could choose to use the number of queries per second as a criterion to measure extremeness. Then using this criterion, we aim to simulate extreme (i.e. rapidly arriving) but realistic query loads for the purpose of stress testing. Another example is rainfall data over a map, as in Figure 1. Here, we are interested in flood resilience, so we can choose to measure extremeness based on total rainfall. Then, generating realistic extreme samples would mean generating rainfall scenarios with spatially realistic patterns that resemble rainfall patterns in actual floods, such as in the right side of Figure 1, which could be used for testing the resilience of a city's flood planning infrastructure.

To model extremeness in a principled way, our approach draws from Extreme Value Theory (EVT), a probabilistic framework designed for modelling the extreme tails of distributions. However, there are two additional aspects to this problem which make it challenging. The first issue is the lack
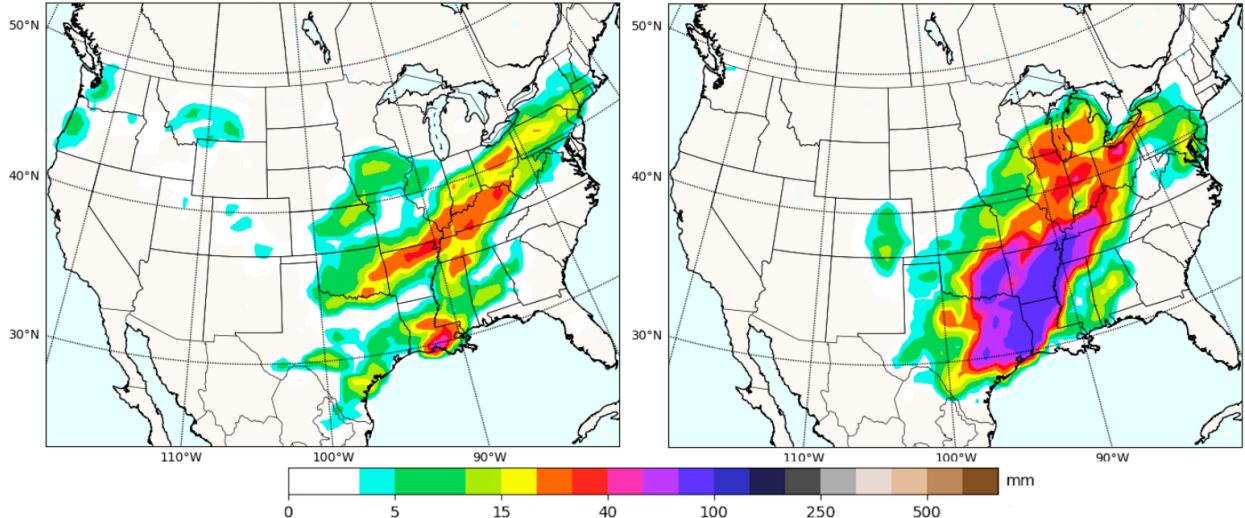
---

\*Equal Contribution

Figure 1: Our goal is to generate samples which are both *realistic* and *extreme*, based on any user-specified extremeness criteria (in this case, high total rainfall). *Left:* Existing GAN-based approaches generate typical rainfall patterns, which have low (green) to moderate (red) rainfall. *Right:* Extreme samples generated by our approach have extreme (violet) rainfall, and realistic spatial patterns resembling that of real floods.

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. The second issue is that we need to generate extreme samples at any given, user-specified extremeness probability.

One possible approach is to train a GAN, say DCGAN (Radford, Metz, and Chintala 2016), over all the images in the dataset regardless of their extremeness. A rejection sampling strategy can then be applied, where images are generated repeatedly until an example satisfying the desired extremeness probability is found. However, as we show in Section 5, the time taken to generate extreme samples increases rapidly with increasing extremeness, resulting in poor scalability.

Our approach, ExGAN, relies on two key ideas. Firstly, to mitigate the lack of training data in the extreme tails of the data distribution, we use a novel **distribution shifting** approach, which gradually shifts the data distribution in the direction of increasing extremeness. This allows us to fit a GAN in a robust and stable manner, while fitting the tail of the distribution, rather than its bulk. Secondly, to generate data at any given extremeness probability, we use **EVT-based conditional generation**: we train a conditional GAN, conditioned on the extremeness statistic. This is combined with EVT analysis, along with keeping track of the amount of distribution shifting performed, to generate new samples at the given extremeness probability.

We present a thorough analysis of our approach, ExGAN, on the US precipitation data. This dataset consists of daily precipitation data over a spatial grid across the lower 48 United States (Continental United States), Puerto Rico, and Alaska. The criteria used to define extremeness is the total rainfall, and, as explained above, an extreme scenario would correspond to a flood. We show that we are able to generate

realistic and extreme rainfall patterns.

Figure 2 shows images of rainfall patterns from the data, both normal and extreme samples, and images sampled from DCGAN and ExGAN simulating normal and extreme conditions.

In summary, the main contributions of our approach are:

1. **Generating Extreme Samples:** We propose a novel deep learning-based approach for generating extreme data using distribution-shifting and EVT analysis.

2. **Constant Time Sampling:** We demonstrate how our approach is able to generate extreme samples in constant-time (with respect to the extremeness probability $\tau$), as opposed to the $\mathcal{O}(\frac{1}{\tau})$ time taken by the baseline approach.

3. **Effectiveness:** Our experimental results show that ExGAN generates realistic samples based on both visual inspection and quantitative metrics, and is faster than the baseline approach by at least three orders of magnitude for extremeness probability of $0.01$ and beyond.

**Reproducibility**: Our code and datasets are publicly available at https://github.com/Stream-AD/ExGAN.

## 2 Related Work
### 2.1 Conditional Generative Adversarial Networks
Conditional GANs (CGANs), introduced in (Mirza and Osindero 2014), allow additional information as input to GAN which makes it possible to direct the data generation process. Conditional DCGAN (CDCGAN) (Gauthier 2015), is a modification of CGAN using the conditional variables but with a convolutional architecture. These methods are discussed briefly in Appendix D. There has also been a significant amount of work done on GAN based models for conditioning on different type of inputs such as images (Zhu et al. 2017;
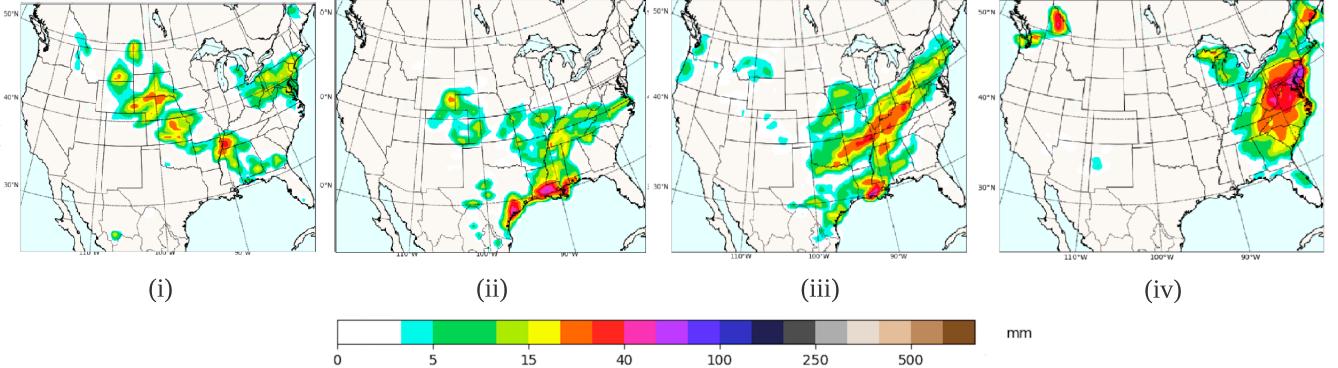
(a) Normal samples ((i) and (ii)) from the original dataset show low and moderate rainfall. Samples generated using DCGAN ((iii) and (iv)) are similar to normal samples from the original dataset.
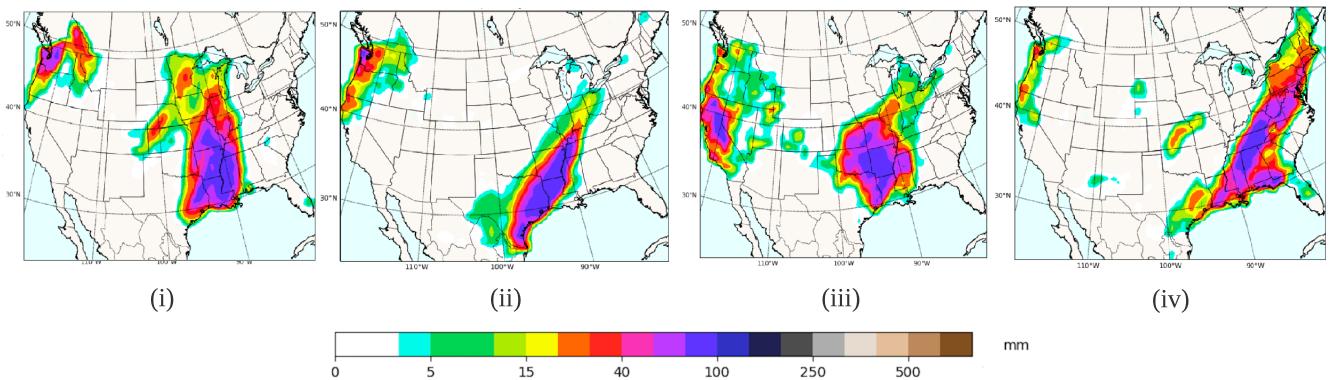


(b) Extreme samples ((i) and (ii)) from the original dataset showing high rainfall. Samples generated using ExGAN ((iii) and (iv)) are similar to extreme samples from the original dataset.

Figure 2: Comparison between DCGAN (which generates normal samples), and ExGAN (which generates extreme samples).

Kim et al. 2017), text (Reed et al. 2016b), and multi-modal conditional GANs (Reed et al. 2016a).

## 2.2 Data Augmentation

Data Augmentation using GANs (Antoniou, Storkey, and Edwards 2017; Shmelkov, Schmid, and Alahari 2018; Tran et al. 2017, 2020; Yamaguchi, Kanai, and Eda 2020; Karras et al. 2020) has been extensively used in different domains, such as anomaly detection (Lim et al. 2018), time series (Zhou et al. 2019; Ramponi et al. 2018), speech processing (Zhang et al. 2019), NLP (Chang, Chuang, and yi Lee 2019; Yu et al. 2017; Fedus, Goodfellow, and Dai 2018), emotion classification (Zhu et al. 2018; Luo and Lu 2018), medical applications (Zheng, Zheng, and Yang 2017; Han et al. 2019; Hu et al. 2018; Calimeri et al. 2017) and computer vision (Karras, Laine, and Aila 2019; Odena, Olah, and Shlens 2017; Perez and Wang 2017; Sixt, Wild, and Landgraf 2018; Choi, Kim, and Kim 2019; Siarohin et al. 2019) as a solution for tackling class imbalance (Mariani et al. 2018) and generating cross-domain data (Huang et al. 2018). However, these methods do not provide any control over the extremeness of the generated data.

## 2.3 Extreme Value Theory

Extreme value theory (Gumbel 2012; Pickands 1975) is a statistical framework for modelling extreme deviations or tails of probability distributions. EVT has been applied to a variety of machine learning tasks including anomaly detection (Guggilam et al. 2019; Siffer et al. 2017; Vignotto and Engelke 2020; Thomas et al. 2017; Goix, Sabourin, and Clémençon 2016), graph mining (Hooi et al. 2020) and local intrinsic dimensionality estimation (Ma et al. 2018; Amsaleg et al. 2018). (Jalalzai, Clémençon, and Sabourin 2018) use EVT to develop a probabilistic framework for classification in extreme regions, (Weng et al. 2018) use it to design an attack-agnostic robustness metric for neural networks.

EVT typically focuses on modelling univariate or low-dimensional (Tawn 1990) distributions. A few approaches, such as dimensionality-reduction based (Chautru 2015; Sabourin and Naveau 2014), exist for moderate dimensional vectors (e.g. 20). A popular approach for multivariate extreme value analysis is Peaks-over-Threshold with specific definitions of exceedances (Rootzén, Tajvidi et al. 2006; Ferreira, De Haan et al. 2014; Engelke et al. 2015), and (Dombry and Ribatet 2015) showed it can be modelled by r-Pareto processes. (de Fondeville and Davison 2016, 2020) presented an

inference method on r-Pareto processes applicable to higher dimensions compared to previous works on max-stable processes (Asadi, Davison, and Engelke 2015) and Pareto processes (Thibaud and Opitz 2015).

To the best of our knowledge, there has not been any work on extreme sample generation using deep generative models.

## 3 Background

### 3.1 Extreme Value Theory (EVT)

The Generalized Pareto Distribution (GPD) (Coles et al. 2001) is a commonly used distribution in EVT. The parameters of GPD are its **scale** $\sigma$, and its **shape** $\xi$. The cumulative distribution function (CDF) of the GPD is:

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

A useful property of the GPD is that it generalizes both Pareto distributions (which have heavy tails) and exponential distributions (which have exponentially decaying tails). In this way, the GPD can model both heavy tails and exponential tails, and smoothly interpolate between them. Another property of the GPD is its 'universality' property for tails: intuitively, it can approximate the tails of a large class of distributions following certain smoothness conditions, with error approaching 0. Thus, the GPD is particularly suitable for modelling the tails of distributions.

(Pickands 1975; Balkema and De Haan 1974) show that the excess over a sufficiently large threshold $u$, denoted by $X - u$, is likely to follow a Generalized Pareto Distribution (GPD) with parameters $\sigma(u), \xi$. This is also known as the Peaks over Threshold method. In practice, the threshold $u$ is commonly set to a value around the $95^{th}$ percentile, while the remaining parameters can be estimated using maximum likelihood estimation (Grimshaw 1993).

## 4 ExGAN: Extreme Sample Generation Using GANs

### 4.1 Problem

We are given a training set $\mathbf{x}_1, \cdots, \mathbf{x}_n \sim \mathcal{D}$, along with $\mathsf{E}(\mathbf{x})$, a user-defined *extremeness measure*: for example, in our running example of rainfall modelling, the extremeness measure is defined as the total rainfall in $\mathbf{x}$, but any measure could be chosen in general. We are also given a user-specified *extremeness probability* $\tau \in (0, 1)$, representing 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.[1]

Given these, our goal is to generate synthetic samples $\mathbf{x}'$ that are both 1) *realistic*, i.e. hard to distinguish from the training data, and 2) *extreme* at the given level: that is, $P_{\mathbf{x} \sim \mathcal{D}}(\mathsf{E}(\mathbf{x}) > \mathsf{E}(\mathbf{x}'))$ should be as close as possible to $\tau$.

[1]In hydrology, the notion of a 100-*year flood* is a well-known concept used for flood planning and regulation, which is defined as a flood that has a 1 in 100 chance of being exceeded in any given year. Given daily data, generating a 100-year flood then corresponds to setting $\tau = \frac{1}{365 \times 100}$.

### 4.2 Distribution Shifting

An immediate issue we face is that we want our trained model to mimic the extreme tails, not the bulk of the distribution; however, most of the data lies in its bulk, with much fewer samples in its tails. While data augmentation could be employed, techniques like image transforms may not be applicable: for example, in the US precipitation data, each pixel captures the rainfall distribution at some fixed location; altering the image using random transforms would change this correspondence.

To address this issue, we propose a novel Distribution Shifting approach in Algorithm 1, parameterized by a shift parameter $c \in (0, 1)$. Our overall approach is to repeatedly 'shift' the distribution by filtering away the less extreme $(1-c)$ proportion of the data, then generating data to return the dataset to its original size. In addition, to maintain the desired proportion of original data points from $\mathcal{X}$, we adopt a 'stratified' filtering approach, where the original and generated data are filtered separately.

Specifically, we first sort our original dataset $\mathcal{X}$ in decreasing order of extremeness (Line 2), then initialize our shifted dataset $\mathcal{X}_s$ as $\mathcal{X}$ (Line 3). Next, each iteration $i$ of a Distribution Shift operation works as follows. We first fit a DCGAN to $\mathcal{X}_s$ (Line 6). We then replace our shifted dataset $\mathcal{X}_s$ with the top $\lfloor c^i \cdot n \rfloor$ extreme data points from $\mathcal{X}$ (Line 7). Next, we use the DCGAN to generate additional $\lceil (n - \lfloor c^i \cdot n \rfloor) \cdot \frac{1}{c} \rceil$ data samples and add the most extreme $n - \lfloor c^i \cdot n \rfloor$ samples to $\mathcal{X}_s$ (Line 8). This ensures that we choose the most extreme $c$ proportion of the generated data, while bringing the dataset back to its original size of $n$ data points. Each such iteration shifts the distribution toward its upper tail by a factor of $c$. We perform $k$ iterations, aiming to shift the distribution sufficiently so that $\tau$ is no longer in the extreme tail of the resulting shifted distribution. Iteratively shifting the distribution in this way ensures that we always have enough data to train the GAN in a stable manner, while allowing us to gradually approach the tails of the distribution.

---

**Algorithm 1:** Distribution Shifting

1 **Input**: dataset $\mathcal{X}$, extremeness measure $\mathsf{E}$, shift parameter $c$, iteration count $k$
2 Sort $\mathcal{X}$ in decreasing order of extremeness
3 Initialize $\mathcal{X}_s \leftarrow \mathcal{X}$
4 **for** $i \leftarrow 1$ *to* $k$ **do**
5    ▷ **Shift the data distribution by a factor of $c$:**
6    Train DCGAN $G$ and $D$ on $\mathcal{X}_s$
7    $\mathcal{X}_s \leftarrow$ top $\lfloor c^i \cdot n \rfloor$ extreme samples of $\mathcal{X}$
8    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 $\mathcal{X}_s$

---

In addition, during the shifting process, we can train successive iterations of the generator via 'warm start', by initializing its parameters using the previous trained model, for the sake of efficiency.

## 4.3 EVT-based Conditional Generation

The next issue we face is the need to generate samples at the user-given extremeness probability of $\tau$. Our approach will be to train a conditional GAN using extremeness as conditioning variable. To generate samples, we then use EVT analysis, along with our knowledge of how much shifting has been performed, to determine the necessary extremeness level we should condition on, to match the desired extremeness probability.

Specifically, first note that after $k$ shifts, the corresponding extremeness probability in the shifted distribution that we need to sample at becomes $\tau' = \tau/c^k$. Thus, it remains to sample from the shifted distribution at the extremeness probability of $\tau'$, which we will do using EVT. Algorithm 2 describes our approach: we first compute the extremeness values using $\mathsf{E}$ on each point in $\mathcal{X}_s$: i.e. $e_i = \mathsf{E}(\mathbf{x_i}) \, \forall \, \mathbf{x_i} \in \mathcal{X}_s$ (Line 2). Then we perform EVT Analysis on $e_1, \cdots, e_n$: we fit Generalized Pareto Distribution (GPD) parameters $\sigma, \xi$ using maximum likelihood estimation (Grimshaw 1993) to $e_1, \cdots, e_n$ (Line 3). Next, we train a conditional DCGAN (Generator $G_s$ and Discriminator $D_s$) on $\mathcal{X}_s$, with the conditional input to $G_s$ (within the training loop of $G_s$) sampled from a GPD with parameters $\sigma, \xi$ (Line 4). In addition to the image, $D_s$ takes in a second input which is $e$ for a generated image $G_s(\mathbf{z}, e)$ and $\mathsf{E}(\mathbf{x})$ for a real image $\mathbf{x}$. An additional loss $\mathcal{L}_{\text{ext}}$ is added to the GAN objective:

$$\mathcal{L}_{\text{ext}} = \mathbb{E}_{\mathbf{z}, e} \left[ \frac{|e - \mathsf{E}(G_s(\mathbf{z}, e))|}{e} \right] \qquad (2)$$

where $\mathbf{z}$ is sampled from multivariate standard normal distribution and $e$ is sampled from a GPD with parameters $\sigma, \xi$. Note that training using $\mathcal{L}_{\text{ext}}$ requires $\mathsf{E}$ to be differentiable.

$\mathcal{L}_{\text{ext}}$ minimizes the distance between the desired extremeness ($e$) and the extremeness of the generated sample ($\mathsf{E}(G_s(z, e))$). This helps reinforce the conditional generation property and prevents the generation of samples with unrelated extremeness. Using the inverse CDF of the GPD, we determine the extremeness level $e'$ that corresponds to an extremeness probability of $\tau'$:

$$e' = G_{\sigma, \xi}^{-1}(1 - \tau') \qquad (3)$$

where $G_{\sigma, \xi}^{-1}$ is the inverse CDF of the fitted GPD (Line 5). Finally, we sample from our conditional DCGAN at the desired extremeness level $e'$ (Line 6).

---

**Algorithm 2:** EVT-based Conditional Generation

1 **Input**: shifted dataset $\mathcal{X}_s$, extremeness measure $\mathsf{E}$, adjusted extremeness probability $\tau'$
2 Compute extremeness values $e_i = \mathsf{E}(\mathbf{x_i}) \, \forall \, \mathbf{x_i} \in \mathcal{X}_s$
3 Fit GPD parameters $\sigma, \xi$ 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 $\sigma, \xi$
5 Extract required extremeness level: $e' \leftarrow G_{\sigma, \xi}^{-1}(1 - \tau')$
6 Sample from $G_s$ conditioned on extremeness level $e'$

---

## 5  Experiments

In this section, we evaluate the performance of ExGAN compared to DCGAN on the US precipitation data. We aim to answer the following questions:

**Q1. Realistic Samples (Visual Inspection):** Does ExGAN generate realistic extreme samples, as evaluated by visual inspection of the images?

**Q2. Realistic Samples (Quantitative Measures):** Does ExGAN generate realistic extreme samples, as evaluated using suitable GAN metrics?

**Q3. Speed:** How fast does ExGAN generate extreme samples compared to the baseline? Does it scale with high extremeness?

Details about our experimental setup, network architecture and software implementation can be found in Appendix A, B and C respectively.

**Dataset:** We use the US precipitation dataset [2]. The National Weather Service employs a multi-sensor approach to calculate the observed precipitation with a spatial resolution of roughly $4 \times 4$ km on an hourly basis. We use the daily spatial rainfall distribution for the duration January 2010 to December 2016 as our training set, and for the duration of January 2017 to August 2020 as our test set. We only retain those samples in our test set which are more extreme, i.e. have higher total rainfall, than the $95^{th}$ percentile in the train set. Images with original size $813 \times 1051$ are resized to $64 \times 64$ and normalized between $-1$ and $1$.

**Baseline:** The baseline is a DCGAN (Radford, Metz, and Chintala 2016) trained over all the images in the dataset, combined with rejection sampling. Specifically, to generate at a user-specified level $\tau$, we use EVT as in our framework (i.e. Eq. (3)) to compute the extremeness level $e = G_{\sigma, \xi}^{-1}(1 - \tau)$ that corresponds to an extremeness probability of $\tau$. We then repeatedly generate images until one is found that satisfies the extremeness criterion within $10\%$ error; that is, we reject the image $\mathbf{x}$ if $\left| \frac{e - \mathsf{E}(\mathbf{x})}{e} \right| > 0.1$.

**Evaluation Metrics:** We evaluate how effectively the generator is able to mimic the tail of the distribution using FID and Reconstruction Loss metrics. Fréchet Inception Distance (FID) (Heusel et al. 2017) is a common metric used in the GAN literature to evaluate image samples and has been found to be consistent with human judgement. Intuitively, it compares the distributions of real and generated samples based on their activation distributions in a pre-trained network. However, an ImageNet-pretrained Inception network which is usually used to calculate FID is not suitable for our dataset. Hence, we construct an autoencoder trained on test data, as described above, and use the statistics on its bottleneck activations to compute the FID:

$$\text{FID} = \|\boldsymbol{\mu_r} - \boldsymbol{\mu_g}\|^2 + \text{Tr}\left( \boldsymbol{\Sigma_r} + \boldsymbol{\Sigma_g} - 2\left(\boldsymbol{\Sigma_r}\boldsymbol{\Sigma_g}\right)^{1/2} \right)$$

---

[2]https://water.weather.gov/precip/

where Tr denotes the trace of a matrix, $(\boldsymbol{\mu_r}, \boldsymbol{\Sigma_r})$ and $(\boldsymbol{\mu_g}, \boldsymbol{\Sigma_g})$ are the mean and covariance of the bottleneck activations for the real and generated samples respectively.

We further evaluate our model on its ability to reconstruct unseen extreme samples by computing a reconstruction loss on the test set (Xiang and Li 2017).

Letting $\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_m$ denote the test images, the reconstruction loss for an unconditional generator $G$ is given by,

$$\mathcal{L}_{\text{rec}} = \frac{1}{m} \sum_{i=1}^{m} \min_{\mathbf{z_i}} \|G(\mathbf{z_i}) - \tilde{\mathbf{x}}_i\|_2^2$$

where $\mathbf{z_i}$ are the latent space vectors

For an extremeness conditioned generator $G$,

$$\mathcal{L}_{\text{rec\_ext}} = \frac{1}{m} \sum_{i=1}^{m} \min_{\mathbf{z_i}} \|G(\mathbf{z_i}, \mathsf{E}(\tilde{\mathbf{x}}_i)) - \tilde{\mathbf{x}}_i\|_2^2$$

To compute the reconstruction loss, we initialize the latent space vectors $\mathbf{z_i}$ as the zero vector, and perform gradient descent on it to minimize the objective defined above. We use similar parameters as (Xiang and Li 2017) to calculate the reconstruction loss, i.e. learning rate was set to 0.001 and number of gradient descent steps was set to 2000, while we use Adam optimizer instead of RMSprop.

We also evaluate how accurately our method is able to condition on the extremeness of the samples. We use Mean Absolute Percentage Error (MAPE), where the error is calculated between the extremeness used to generate the sample ($e$) and the extremeness of the generated sample ($\mathsf{E}(G(\mathbf{z}, e))$).

$$\text{MAPE} = \mathbb{E}_{\mathbf{z}, e} \left[ \frac{|e - \mathsf{E}(G_s(\mathbf{z}, e))|}{e} \right] \times 100\% \qquad (4)$$

where $\mathbf{z}$ is sampled from multivariate standard normal distribution and $e$ is sampled from a GPD with parameters $\sigma, \xi$.

## 5.1 Realistic Samples (Visual Inspection)

Figure 3 shows the extreme samples generated by ExGAN corresponding to extremeness probability $\tau = 0.001$ and $0.0001$. We observe that ExGAN generates samples that are similar to the images of rainfall patterns from the original data in Figure 2b. As we change $\tau$ from 0.001 to 0.0001, we observe the increasing precipitation in the generated samples. The typical pattern of radially decreasing rainfall in real data is learned by ExGAN. ExGAN also learns that coastal areas are more susceptible to heavy rainfall.

Figure 3c shows the extreme samples generated by DCGAN for extremeness probability $\tau = 0.01$. When $\tau = 0.001$ or $0.0001$, DCGAN is unable to generate even one sample, within 10% error, in 1 hour (as we explain further in Section 5.3).

## 5.2 Realistic Samples (Quantitative Measures)

The GAN is trained for 100 epochs in each iteration of distribution shifting. For distribution shifting, we set $c = 0.75$, $k = 10$ and use warm start. MAPE for DCGAN can be upper bounded by the rejection strategy used for sampling, and this bound can be made tighter at the expense of sampling

time. For our experiment, we upper bound the MAPE for DCGAN by 10% as explained above. MAPE for ExGAN is $3.14\% \pm 3.08\%$.

Table 1 reports the FID (lower is better) and reconstruction loss (lower is better). ExGAN is able to capture the structure and extremeness in the data, and generalizes better to unseen extreme scenarios, as shown by the lower reconstruction loss and lower FID score (loss = 0.0172 and FID = $0.0236 \pm 0.0037$) as compared to DCGAN (loss = 0.0292 and FID = $0.0406 \pm 0.0063$).

| Method | FID | Reconstruction Loss |
|---|---|---|
| DCGAN | $0.0406 \pm 0.0063$ | 0.0292 |
| ExGAN | $0.0236 \pm 0.0037$ | 0.0172 |

Table 1: FID, and Reconstruction Loss, for DCGAN and ExGAN (averaged over 5 runs). For FID, the p-value for significant improvement of ExGAN over the baseline is 0.002, using a standard two-sample t-test.

Table 2 reports the reconstruction loss, MAPE and FID for ExGAN for different values of $c$ and $k$. To ensure a fair comparison, we select the parameters $c$ and $k$ for distribution shifting, such that the amount of shift, $c^k$, is approximately similar. Intuitively, we would expect higher $c$ to correspond to slower and more gradual shifting, which in turn helps the network smoothly interpolate and adapt to the shifted distribution, leading to better performance. This trend is observed in Table 2. However, these performance gains with higher $c$ values come at the cost of training time.

In Appendix E, we report ablation results on distribution shifting illustrating its benefit to our approach.

| $c$ | $k$ | Rec. Loss | MAPE | FID |
|---|---|---|---|---|
| 0.24 | 2 | 0.0173 | $3.43 \pm 3.01$ | $0.0367 \pm 0.0096$ |
| 0.49 | 4 | 0.0173 | $3.32 \pm 3.10$ | $0.0304 \pm 0.0109$ |
| 0.75 | 10 | 0.0172 | $3.14 \pm 3.08$ | $0.0236 \pm 0.0037$ |
| 0.90 | 27 | 0.0169 | $3.05 \pm 3.14$ | $0.0223 \pm 0.0121$ |

Table 2: Reconstruction Loss, MAPE and FID values for ExGAN for different $c$ and $k$ (averaged over 5 runs).

## 5.3 Speed

The time taken to generate 100 samples for different extremeness probabilities is reported in Table 3. Note that ExGAN is scalable and generates extreme samples in constant time as opposed to the $\mathcal{O}(\frac{1}{\tau})$ time taken by DCGAN to generate samples with extremeness probability $\tau$. DCGAN could not generate even one sample for extremeness probabilities $\tau = 0.001$ and $\tau = 0.0001$ in 1 hour. Hence, we do not report sampling times on DCGAN for these two values.
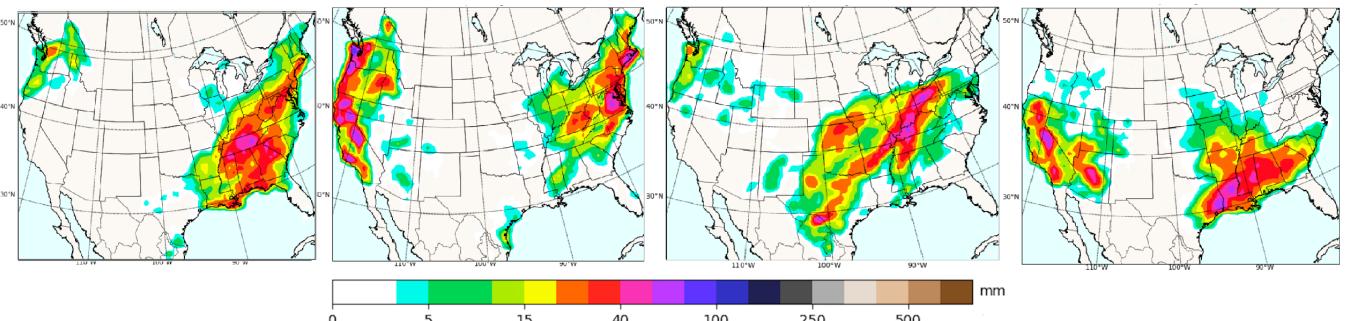
## 6 Conclusion

In this paper, we propose ExGAN, a novel deep learning-based approach for generating extreme data. We use (a) dis-
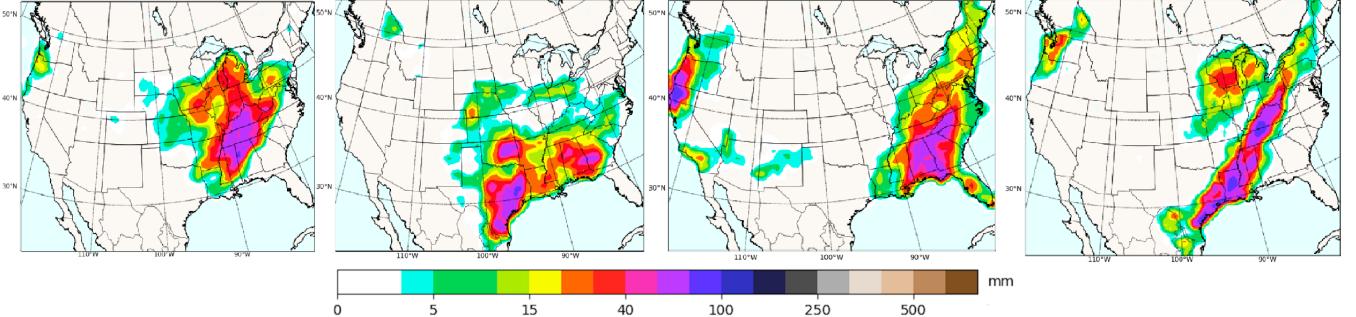
(a) Samples from ExGAN for extremeness probability $\tau = 0.001$. Time taken to sample $= 0.002s$
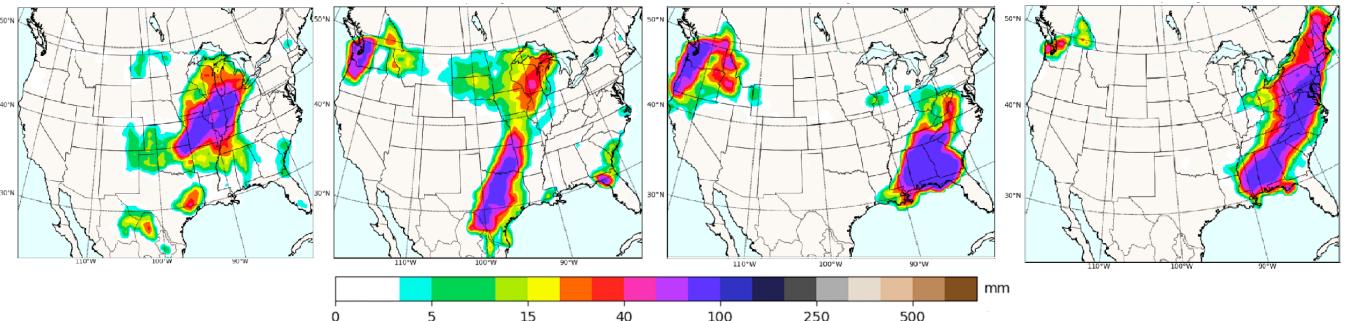


(b) Samples from ExGAN for extremeness probability $\tau = 0.0001$. Time taken to sample $= 0.002s$



(c) Samples from DCGAN for extremeness probability $\tau = 0.01$. Time taken to sample $= 7.564s$. DCGAN is unable to generate samples in 1 hour when $\tau = 0.001$ or $0.0001$.

Figure 3: ExGAN generates images which are realistic, similar to the original data samples, in constant time.

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

Table 3: Sampling times for DCGAN and ExGAN for different extremeness probabilities (in seconds).

tribution shifting to mitigate the lack of training data in the extreme tails of the data distribution; (b) EVT-based conditional generation to generate data at any given extremeness probability.

We demonstrate how our approach is able to generate extreme samples in constant-time (with respect to the ex-

tremeness probability $\tau$), as opposed to the $\mathcal{O}(\frac{1}{\tau})$ time taken by the baseline. Our experimental results show that ExGAN generates realistic samples based on both visual inspection and quantitative metrics, and is faster than the baseline approach by at least three orders of magnitude for extremeness probability of $0.01$ and beyond.

The flexibility and realism achieved by the inclusion of GANs, however, comes at the cost of theoretical guarantees. While our algorithmic steps (e.g. Distribution Shifting) are designed to approximate the tails of the original distribution in a principled way, it is difficult to provide guarantees due to its GAN framework. Future work could consider different model families (e.g. Bayesian models), toward the goal of deriving theoretical guarantees, as well as incorporating neural network based function approximators to learn a suitable extremeness measure (E).

## Ethical Impact

Modelling extreme events in order to evaluate and mitigate their risk is a fundamental goal in a wide range of applications, such as extreme weather events, financial crashes, and managing unexpectedly high demand for online services. Our method aims to generate realistic and extreme samples at any user-specified probability level, for the purpose of planning against extreme scenarios, as well as stress-testing of existing systems. Our work also relates to the goal of designing robust and reliable algorithms for safety-critical applications such as medical applications, aircraft control, and many others, by exploring how we can understand and generate the extremes of a distribution.

Our work explores the use of deep generative models for generating realistic extreme samples, toward the goal of building robust and reliable systems. Possible negative impact can arise if these samples are not truly representative or realistic enough, or do not cover a comprehensive range of possible extreme cases. Hence, more research is needed, such as for ensuring certifiability or verifiability, as well as evaluating the practical reliability of our approach for stress-testing in a wider range of real-world settings.

## References

Amsaleg, L.; Chelly, O.; Furon, T.; Girard, S.; Houle, M. E.; Kawarabayashi, K.-i.; and Nett, M. 2018. Extreme-value-theoretic estimation of local intrinsic dimensionality. *Data Mining and Knowledge Discovery* .

Antoniou, A.; Storkey, A.; and Edwards, H. 2017. Data Augmentation Generative Adversarial Networks. *ICLR* .

Asadi, P.; Davison, A. C.; and Engelke, S. 2015. Extremes on river networks. *The Annals of Applied Statistics* .

Balkema, A. A.; and De Haan, L. 1974. Residual Life Time at Great Age. In *The Annals of probability*.

Calimeri, F.; Marzullo, A.; Stamile, C.; and Terracina, G. 2017. Biomedical Data Augmentation Using Generative Adversarial Neural Networks. In *ICANN*.

Chang, C.-T.; Chuang, S.-P.; and yi Lee, H. 2019. Code-switching Sentence Generation by Generative Adversarial Networks and its Application to Data Augmentation. In *INTERSPEECH*.

Chautru, E. 2015. Dimension reduction in multivariate extreme value analysis. *Electronic Journal of Statistics* .

Choi, J.; Kim, T.-K.; and Kim, C. 2019. Self-Ensembling With GAN-Based Data Augmentation for Domain Adaptation in Semantic Segmentation. *ICCV* .

Coles, S.; Bawa, J.; Trenner, L.; and Dorazio, P. 2001. An Introduction to Statistical Modeling of Extreme Values. *JASA* .

de Fondeville, R.; and Davison, A. C. 2016. High-dimensional peaks-over-threshold inference. *Biometrika* .

de Fondeville, R.; and Davison, A. C. 2020. Functional Peaks-over-threshold Analysis. *ArXiv* abs/2002.02711.

Dombry, C.; and Ribatet, M. 2015. Functional regular variations, Pareto processes and peaks over threshold. *Statistics and Its Interface* .

Engelke, S.; Malinowski, A.; Kabluchko, Z.; and Schlather, M. 2015. Estimation of hüsler–reiss distributions and brown–resnick processes. *Statistical Methodology* .

Fedus, W.; Goodfellow, I.; and Dai, A. M. 2018. MaskGAN:Better Text Generation via Filling in the _. *ICLR* .

Ferreira, A.; De Haan, L.; et al. 2014. The generalized Pareto process; with a view towards application and simulation. *Bernoulli* .

Gauthier, J. 2015. Conditional generative adversarial nets for convolutional face generation. In *Stanford CS231N class project*.

Goix, N.; Sabourin, A.; and Clémençon, S. 2016. Sparse representation of multivariate extremes with applications to anomaly ranking. In *AISTATS*.

Grimshaw, S. D. 1993. Computing maximum likelihood estimates for the generalized Pareto distribution. *Technometrics* .

Guggilam, S.; Zaidi, S. M. A.; Chandola, V.; and Patra, A. K. 2019. Bayesian Anomaly Detection Using Extreme Value Theory. *ArXiv* abs/1905.12150.

Gumbel, E. J. 2012. *Statistics of extremes*. Courier Corporation.

Han, C.; Murao, K.; Noguchi, T.; Kawata, Y.; Uchiyama, F.; Rundo, L.; Nakayama, H.; and Satoh, S. 2019. Learning more with less: Conditional PGGAN-based data augmentation for brain metastases detection using highly-rough annotation on MR images. In *CIKM*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NIPS*.

Hooi, B.; Shin, K.; Lamba, H.; and Faloutsos, C. 2020. Tell-Tail: Fast Scoring and Detection of Dense Subgraphs. In *AAAI*.

Hu, X.; Chung, A. G.; Fieguth, P.; Khalvati, F.; Haider, M. A.; and Wong, A. 2018. ProstateGAN: Mitigating Data Bias via Prostate Diffusion Imaging Synthesis with Generative Adversarial Networks. *ArXiv* abs/1811.05817.

Huang, S.-W.; Lin, C.-T.; Chen, S.-P.; Wu, Y.-Y.; Hsu, P.-H.; and Lai, S.-H. 2018. AugGAN: Cross Domain Adaptation with GAN-Based Data Augmentation. In *ECCV*.

Jalalzai, H.; Clémençon, S.; and Sabourin, A. 2018. On Binary Classification in Extreme Regions. In *NeurIPS*.

Karras, T.; Aittala, M.; Hellsten, J.; Laine, S.; Lehtinen, J.; and Aila, T. 2020. Training generative adversarial networks with limited data. *NeurIPS* .

Karras, T.; Laine, S.; and Aila, T. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. *CVPR* .

Kim, T.; Cha, M.; Kim, H.; Lee, J. K.; and Kim, J. 2017. Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. In *ICML*.

Lim, S. K.; Loo, Y.; Tran, N.-T.; Cheung, N.-M.; Roig, G.; and Elovici, Y. 2018. DOPING: Generative Data Augmentation for Unsupervised Anomaly Detection with GAN. *ICDM* .

Luo, Y.; and Lu, B.-L. 2018. EEG data augmentation for emotion recognition using a conditional wasserstein GAN. In *EMBC*.

Ma, X.; Li, B.; Wang, Y.; Erfani, S. M.; Wijewickrema, S.; Schoenebeck, G.; Song, D.; Houle, M. E.; and Bailey, J. 2018. Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality. *ICLR* .

Mariani, G.; Scheidegger, F.; Istrate, R.; Bekas, C.; and Malossi, C. 2018. BAGAN: Data Augmentation with Balancing GAN. *ArXiv* abs/1803.09655.

Mirza, M.; and Osindero, S. 2014. Conditional Generative Adversarial Nets. *ArXiv* abs/1411.1784.

Odena, A.; Olah, C.; and Shlens, J. 2017. Conditional image synthesis with auxiliary classifier gans. In *ICML.*

Perez, L.; and Wang, J. 2017. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *ArXiv* abs/1712.04621.

Pickands, J. 1975. Statistical Inference Using Extreme Order Statistics. In *Annals of statistics*.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR* .

Ramponi, G.; Protopapas, P.; Brambilla, M.; and Janssen, R. 2018. T-CGAN: Conditional Generative Adversarial Network for Data Augmentation in Noisy Time Series with Irregular Sampling. *ArXiv* abs/1811.08295.

Reed, S.; Akata, Z.; Mohan, S.; Tenka, S.; Schiele, B.; and Lee, H. 2016a. Learning What and Where to Draw. In *NIPS*.

Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; and Lee, H. 2016b. Generative Adversarial Text to Image Synthesis. *ICML* .

Rootzén, H.; Tajvidi, N.; et al. 2006. Multivariate generalized Pareto distributions. *Bernoulli* .

Sabourin, A.; and Naveau, P. 2014. Bayesian Dirichlet mixture model for multivariate extremes: A re-parametrization. *Computational Statistics & Data Analysis* .

Shmelkov, K.; Schmid, C.; and Alahari, K. 2018. How good is my GAN? In *ECCV*.

Siarohin, A.; Lathuilière, S.; Sangineto, E.; and Sebe, N. 2019. Appearance and Pose-Conditioned Human Image Generation using Deformable GANs. *IEEE TPAMI* .

Siffer, A.; Fouque, P.-A.; Termier, A.; Largouet, C.; and Largouët, C. 2017. Anomaly detection in streams with extreme value theory. *KDD* .

Sixt, L.; Wild, B.; and Landgraf, T. 2018. RenderGAN: Generating Realistic Labeled Data. *Frontiers in Robotics and AI* .

Tawn, J. A. 1990. Modelling multivariate extreme value distributions. *Biometrika* .

Thibaud, E.; and Opitz, T. 2015. Efficient inference and simulation for elliptical Pareto processes. *Biometrika* .

Thomas, A.; Clémençon, S.; Gramfort, A.; and Sabourin, A. 2017. Anomaly Detection in Extreme Regions via Empirical MV-sets on the Sphere. In *AISTATS*.

Tran, N.-T.; Tran, V.-H.; Nguyen, N.-B.; Nguyen, T.-K.; and Cheung, N. 2020. Towards Good Practices for Data Augmentation in GAN Training. *ArXiv* abs/2006.05338.

Tran, T.; Pham, T.; Carneiro, G.; Palmer, L.; and Reid, I. 2017. A bayesian data augmentation approach for learning deep models. In *NIPS*.

Vignotto, E.; and Engelke, S. 2020. Extreme value theory for anomaly detection – the GPD classifier. *Extremes* .

Weng, T.-W.; Zhang, H.; Chen, P.-Y.; Yi, J.; Su, D.; Gao, Y.; Hsieh, C.-J.; and Daniel, L. 2018. Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach. *ICLR* .

Xiang, S.; and Li, H. 2017. On the Effects of Batch and Weight Normalization in Generative Adversarial Networks. *ArXiv* abs/1704.03971.

Yamaguchi, S.; Kanai, S.; and Eda, T. 2020. Effective Data Augmentation with Multi-Domain Learning GANs. In *AAAI*.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *AAAI*.

Zhang, X.; Wang, Z.; Liu, D.; and Ling, Q. 2019. DADA: Deep Adversarial Data Augmentation for Extremely Low Data Regime Classification. *ICASSP* .

Zheng, Z.; Zheng, L.; and Yang, Y. 2017. Unlabeled Samples Generated by GAN Improve the Person Re-identification Baseline in Vitro. *ICCV* .

Zhou, B.; Liu, S.; Hooi, B.; Cheng, X.; and Ye, J. 2019. Beat-GAN: Anomalous Rhythm Detection using Adversarially Generated Time Series. In *IJCAI*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Un-paired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *ICCV* .

Zhu, X.; Liu, Y.; Li, J.; Wan, T.; and Qin, Z. 2018. Emotion classification with data augmentation using generative adversarial networks. In *PAKDD*.