# Deep generative modeling of multivariate dependent extremes

Stéphane Girard, Emmanuel Gobet, Jean Pachebat

# Deep generative modeling of multivariate dependent extremes

Stéphane Girard[(1)], Emmanuel Gobet[(2,⋆)], Jean Pachebat[(2)]

[(1)] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France.
[(2)] CMAP, CNRS, Ecole Polytechnique, Institut Polytechnique de Paris,
Route de Saclay, 91128 Palaiseau, France.
[(⋆)] Corresponding author: emmanuel.gobet@polytechnique.edu

**Abstract**

Dealing with extreme values is a major challenge in probabilistic modeling, important in applications such as economics, engineering and life sciences. Models based on transformations of light-tailed noise, such as GANs, fail to capture the tail behaviour of heavy-tailed distributions. In particular, they fail at capturing dependence in extreme regions. We study a modified version of GANs with heavy-tailed input distribution (called HTGAN). Recalling the stable tail dependence function (stdf), a tool from extreme-value theory measuring dependence in extreme regions, we provide a worst-case error bound on the approximation of the stdf of the target with the output of a HTGAN. This bound scales as $N^{-1/(d-1)}$, where $N$ is the dimension of the input noise of the network and $d$ is the dimension of the data. This suggests increasing the dimension of the latent noise to gain precision in the estimation of dependence. We perform experiments, comparing HTGAN with a classical light-tailed GAN (LTGAN) on both synthetic and real datatets exhibiting heavy-tailed characteristics. These experiments confirm our theoretical findings: First, HTGAN is better at reproducing dependence in extremes than LTGAN. Second, the quality of approximation gets better as the dimension of the latent noise increases.

**Keywords:** Generative modeling, GANs, dependence, extremes, heavy-tailed distributions

**MSC2020:** 68T07, 60G70, 62G32, 65C05

## 1 Introduction

**Statement of the problem.** Examining extreme events is a critical concern across various fields such as economics, engineering, and life sciences, with wide-ranging applications like actuarial and financial risks [AA10, Chapters X and XIII]-[EKM97, Chapters 1 and 6]-[MFE15, Chapters 5 and 16, Part III], communication network reliability [Rob03], and aircraft safety [PW05]. Extreme events play a crucial role also in the context of climate change [ZMW+20], with the occurrence of more and more severe weather events, or in the context of cybersecurity [CSF+22] with the increasing number of cyber-attacks of private companies or public entities. In recent decades, the importance of extreme event analysis has surged in financial risk management, which calls for an ever-increasing number of types of risk (market, credit, operational, reputational, cyber, climate, and so on and so forth) to be encompassed in order to measure their impacts on banking activity and the stability of the financial system. In particular, regulators [Eur14] are increasingly imposing stress tests to test the resilience of the banking system against different risk categories. These stress tests involve numerical

simulations of unfavorable yet plausible extreme scenarios, serving as a primary means to evaluate the potential impact on risks (see [ECB23] for the 2023 exercises in Europe). Often, extreme events are described as tail events of multivariate relevant random variables.
The challenge arises in efficiently obtaining relevant samples for assessing these tail risks. Informally, it writes as finding a generator $G$ and a latent distribution $\mu_Z$ which one can easily sample from, such that:

$$Z \sim \mu_Z, \qquad G(Z) \overset{d}{=} \text{target distribution}.$$

A large family of numerical approaches relies on physics-based methods, and aims to efficiently sample the distribution tail with minimal computational effort and to avoid the inefficiency of a basic accept-rejection algorithm. These methods are specific to the model at hand: among these methods, we mention importance sampling [Buc04, Chapter 4], MCMC with splitting – [GL15], or interacting particles system – [DG05]. Another family of simulation schemes is data-driven approaches which do not require the knowledge of the (supposedly existing) physical model behind the sample generation. These data-driven schemes take as input a data set of observations of given size $n_{\text{data}}$ and design a generative model able to re-generate samples that mimic the empirical distribution of the observed data. This concept aligns with recent paradigms in Artificial Intelligence such as Generative Adversarial Networks (GANs) initiated by [GPAM+14] and Variational Autoencoders (VAEs) by [KW14], score-based diffusion models [SDWMG15], normalising flows [PNR+21], etc.
Our work is in this vein for generating extreme samples. Unlike the classical uses of Deep Generative Models as mentioned above, where the number $n_{\text{data}}$ of training data is generally colossal (millions), in the context of extreme events, $n_{\text{data}}$ is small by nature (a few hundred at most). Standard training procedures are thus bound to have poor performance all the more so as, by construction, their design makes them unable to reproduce heavy and dependent tails [AGG22]. The aim of this work is to design new efficient procedures taking advantage of the probabilistic structure behind extreme values.

**State of the art.** Very recently, the machine learning community became aware of the difficulties of sampling extremes using Deep Generative Models, requiring to appropriately adapt known generative methods to the extreme setting. Most of the newly designed algorithms were based on GANs, see an overview in [AGG24]. Three directions have been mainly investigated in the litteratre. First, certain works apply some preprocessing to the data to get rid of the tail heaviness, see Quant-GAN [WKKK20] and evtGAN [BZV+22]: essentially, it consists in suitably transforming each data coordinate in order to retrieve either Gaussian or uniform marginal distributions, then performing a usual GAN method and reverting to the original space by inverse transform. A second direction is to consider new latent variables $Z$ with a heavy-tailed distribution. In [FBS20, HCL+21], a Generalized Pareto Distribution is adopted for the latent distribution $\mu_Z$, in contrast to original GANs which use light-tailed distributions such as uniform or Gaussian. This idea finds its origins in a crucial observation: neural networks are Lipschitz functions [WKKK20]. As such, they cannot transform a random variable with light-tailed to a distribution with heavier tail, this is the main result of [LMH22]. Using heavy-tailed noise however raises an issue with the training procedure: if the underlying distribution is too heavy-tailed, the gradient of the loss function may not exist. To alleviate this, in [HCL+21], the authors introduce a loss function designed to produce stable training. To be effective, these methods require to estimate accurately the tail-index parameters for each marginal distribution, which is a difficult task because the number $n_{\text{data}}$ of training data

is usually small. In [HCL+21], the authors provide empirical evidence that the Pareto-GAN method is able to match heavy-tailed distributions embedded in lower dimensional manifolds in a high dimensional setup. However, they do not provide a quantification of the power of Pareto-GAN to reproduce asymptotic dependence in the data, which is our main scope of interest in this paper.

A third approach consists in suitably parameterize the generator to account for extreme-value theory. This approach has been developed in [AGG22] (EV-GAN) where the renormalized log-quantile (called tail index function by the authors) is learnt using ReLU neural networks. The authors provide error bounds in uniform norm according to the complexity of neural networks and the second-order conditions in extreme-value theory. The latter reference shows that EV-GAN largely outperforms usual GANs, and is able to accurately sample $X$ in dimension up to 50. For higher dimension, the marginals of $X$ are still accurately reproduced but the tail dependence structure is under-estimated. Our work tackles the problem of better learning the dependence between margins with neural networks.

The closest work to ours is presumably [HEN+22]. The authors endeavour to approximate the stable tail dependence function (stdf) of the vector $X$ of interest, see Definition 2.1 later, which characterizes the dependence in the extremes, once the marginals have been standardized to a unit Fréchet scale. Recall that, once the stdf is known (even approximately), the vector $X$ can be sampled using the spectral representation of a max-stable process [HHP18, Theorem 2.1]. In [HEN+22] two learning techniques of the stdf are introduced. On the one hand, a deterministic approach is proposed using a dedicated architecture of neural networks called "d-Max NN". The Fréchet marginalised data are projected onto directions in the simplex leading to exponentially distributed data. An algorithm is then introduced to maximise the log-likelihood using randomised directions on the neural network parameters. On the other hand, a stochastic approach is designed, based on the stochastic representation of the stdf via the spectral measure (see Proposition 2.2). Note that this approach is calibrated using the previous deterministic method. At first glance, our approach may look similar to [HEN+22] but it is actually quite different, even though we also work on unitary Fréchet marginals. Here, we do rely on the spectral representation of max-stable processes, but we rather directly design a generative model of the vector $X$ using an unitary Fréchet latent noise. In our opinion, this is much simpler since our approach avoids truncating (at a poorly controlled rank) the max-stable representation. Then, in contrast to Pareto-GAN, we show a convergence result of the approximated stdf to the true one when the dimension of the latent noise increases.

Last, we shall mention the Tail-GAN of [CCXZ22] in a financial setting, where a usual GAN is performed but the loss function for learning depends on the risk-metrics of the scalar quantity of interest (e.g. Value-at-Risk and Expected Shorfall). It is a way to enforce the generative model to reproduce some specific metrics: note that this is different to sample a multidimensional extreme distribution.

**Our contributions.** We investigate the representation power of neural network based generative models with heavy-tailed input noise. In particular, we study the ability of such models to reproduce multivariate asymptotic dependence, which is quantified by the stdf. To this end, we first provide a modification to the original GAN training to accommodate the specificity of heavy-tailed input noise. Then, the quality of approximation of the stdf by the modified GAN is assessed thanks to the stdf representation in terms of D-norms (see Section 2.2). We show that the approximation error is decreasing when the dimension of the latent noise increases (see Theorem 2.12). Finally, we provide a set of experiments to illustrate the behaviour of

the algorithm and perform a comparison with a benchmark method on both synthetic and real datasets.

**Organization of the paper.** Section 2 is devoted to the theory on the representation capability of the stdf with GANs based on heavy-tailed input noise. The main theoretical tools are introduced before the statement of our approximation results. Experiments on synthetic and real datasets are summarized in Section 3 to illustrate the generative power of heavy-tailed GANs. We provide additional material in the Appendix, specifically: basic results from extreme-value, D-norms and quantization theories, copula tools and measures of dependence used in the paper, as well as the proofs of theoretical results.

**Notations.**

- $[n] = \{i \in \mathbb{N} : \ 1 \leq i \leq n\}$ denotes the set of natural numbers from 1 to $n$. $\lceil \cdot \rceil$ denotes the ceil function.
- The vectors of the canonical basis of $\mathbb{R}^d$ are $\boldsymbol{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0), \ i \in [d]$, with a 1 on the $i^{th}$ coordinate.
- Vectors or matrices are denoted with bold symbols and scalar with roman ones. The $i^{th}$ coordinate of a vector $\boldsymbol{x}$ is referred to as $x_i$. Random variables are denoted with capital letters and constant (deterministic) values with small letters. To fix ideas, $x$ is a constant scalar, $X$ is an $\mathbb{R}$-valued random variable, $\boldsymbol{x}$ is a constant vector and $\boldsymbol{X}$ is a random vector. Vectors with equal scalar coordinates are denoted in bold, such as $\mathbf{0} = (0, \ldots, 0)$ and $\mathbf{1} = (1, \ldots, 1)$. The dimension will be clear from the context.
- Without further specifications, operations such as multiplication, division, max, exponentiation and boolean operations on vectors are meant componentwise. For instance, for two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, $\boldsymbol{x}\,\boldsymbol{y} := (x_1 y_1, \cdots, x_d y_d)$. Boolean operations are meant componentwise: $\boldsymbol{x} \leq \boldsymbol{y} \Leftrightarrow \forall i \in [d], \ x_i \leq y_i$. The scalar product is denoted with a dot: $\boldsymbol{x} \cdot \boldsymbol{y} = \sum_{i=1}^{d} x_i y_i$.
- The cumulative distribution function (cdf) of a random variable $\boldsymbol{X}$ is denoted by $F_{\boldsymbol{X}}$ or $F$ according to the context. The cdf of the extreme-value distribution is denoted by $G$ and the associated domain of attraction is denoted by $\mathrm{Dom}\,(G)$.
- $\mathrm{supp}(P)$ denotes the support of $P$.
- The Pareto distribution is defined by its cdf $1_{x \geq 1}\left(1 - x^{-1/\gamma}\right)$ with $\gamma > 0$. When introducing a Pareto distribution, we might also parametrize it with $\alpha = 1/\gamma$.
- $\#A$ denotes the cardinal of a set $A$.

## 2 Reproducing dependence in extreme regions with Generative Adversarial Networks: Theory

Some background on extreme-value theory is briefly given in Section 2.1. Section 2.2 provides the tools necessary to study dependence in extreme regions of a distribution. Finally, some key elements on Generative Adversarial Networks (GANs) are recalled in Section 2.3. Additional material on each topic is also given in the Appendix. Finally, our theoretical results are

presented in Section 2.4: we provide a quantization bound on how well GANs with heavy-tailed noise (HTGANs) approximate the stdf of a target distribution.

## 2.1 Extreme-value theory

Extreme-value theory is a branch of statistics concerned with studying the upper tails of probability distributions. We refer to Section A.1 in the Appendix for a brief introduction to this theory. Given a $\mathbb{R}^d$ valued random vector $\boldsymbol{X}$ with associated cdf $\boldsymbol{x} \in \mathbb{R}^d \mapsto F(\boldsymbol{x}) = \mathbb{P}(\boldsymbol{X} \leq \boldsymbol{x})$, extreme-value theory establishes the asymptotic distribution of normalized maxima of realizations of $\boldsymbol{X}$. Whence normalized to unit Fréchet margins (A.17), the limiting distribution function $G_\star$ is a simple max-stable distribution, see Definition A.2 and Equation (A.16). As illustrated in the next paragraph, $G_\star$ encodes the dependence structure in the tails of $F$.

## 2.2 Measuring dependence in extremes

We first recall the definition of the stable tail dependence function (stdf) associated with $F$.

**Definition 2.1.** [HF06, Corollary 6.1.4] Let $F$ be a continuous cdf on $\mathbb{R}^d$ belonging to the max-domain of attraction of some distribution. Then, one can define

$$\forall \boldsymbol{x} \in (0,\infty)^d : \ \ell_F(\boldsymbol{x}) = \lim_{t \to +\infty} t\, \mathbb{P}\left(1 - F_1(X_1) \leq \frac{x_1}{t} \ \text{ or } \ \ldots \ \text{ or } \ 1 - F_d(X_d) \leq \frac{x_d}{t}\right),$$

where $F_j$'s are the marginal cdf of $\boldsymbol{X}$. The limit $\ell_F$ is the so-called *stable tail dependence function* (stdf) of $F$.

The stdf provides a "normalized" representation of the dependence structure of the distribution in the tails (as opposed to the copula function which globally models the dependence), in the sense that its marginals are normalized: the stdf does not carry any information on the marginal distributions of the base distribution of interest.
The following proposition gives the classical parametrization of the dependence structure in terms of spectral measure. A more recent and less known parametrization using D-norms is provided in Theorem 2.4 below.

**Proposition 2.2** (Spectral representation of the stdf, [HF06, Remark 6.1.16.])**.** *Consider a cdf $F$ (as in Definition 2.1) and its stdf $\ell_F$. Then, there exists a probability measure $\Lambda$ on the simplex $\Delta_{d-1} := \left\{\boldsymbol{\omega} \geq \boldsymbol{0} : \ \sum_{i\in[d]} \omega_i = 1\right\}$ such that:*

$$\forall \boldsymbol{x} \in (0,\infty)^d : \ \ell_F(\boldsymbol{x}) = d \int_{\Delta_{d-1}} \max_{i\in[d]}(\omega_i x_i) \Lambda(\mathrm{d}\boldsymbol{\omega}), \tag{2.1}$$

*with constraints:*

$$\forall i \in [d] : \ \int_{\Delta_{d-1}} \omega_i \ \Lambda(\mathrm{d}\boldsymbol{\omega}) = 1/d.$$

**Quantifying dependence in extremes with D-norms.** D-norms [Fal19] are a class of norms in vector space, very convenient for studying dependence in extremes.

**Definition 2.3** (D-norms, [Fal19, Lemma 1.1.3])**.** Let $\boldsymbol{\Gamma} = (\boldsymbol{\Gamma}_1, \ldots, \boldsymbol{\Gamma}_d) \in \mathbb{R}^d$ be a random vector, whose components satisfy for all $i \in [d]$, $\boldsymbol{\Gamma}_i \geq 0$ *a.s.* and $\mathbb{E}[\boldsymbol{\Gamma}_i] = 1$. Then,

$$\|\cdot\|_{\boldsymbol{\Gamma}} : \ \boldsymbol{x} \mapsto \|\boldsymbol{x}\|_{\boldsymbol{\Gamma}} = \mathbb{E}\left[\max_{i\in[d]}(|x_i|\boldsymbol{\Gamma}_i)\right] \tag{2.2}$$

defines a norm on $\mathbb{R}^d$, called a D-norm, and $\boldsymbol{\Gamma}$ is called a generator of the D-norm.

Main properties of the D-norm are recalled in Appendix A.2. To alleviate the role of the generator, we may simply denote the above norm by $\|\cdot\|_D$. The key role of D-norms is highlighted in the next result.

**Theorem 2.4.** *Let $F$ be a continuous cdf on $\mathbb{R}^d$ belonging to the max-domain of some distribution. Then, there exists a D-norm that exactly represents the stdf of $F$:*

$$\forall \boldsymbol{x} \in (0,\infty)^d, \qquad \ell_F(\boldsymbol{x}) = \|\boldsymbol{x}\|_D \,. \tag{2.3}$$

This is a consequence of Theorem A.7 in the Appendix. Note that the two representations in (2.1) and in (2.2)–(2.3) may look similar at first sight, but actually they are substantially different. While both representations involve a probability measure (the spectral measure $\Lambda$ on the one hand, and that of the generator $\mathbf{\Gamma}$ associated with the D-norm on the other hand), the second representation is more tractable for approximation purposes: Instead of designing a non-negative measure on the simplex $\Delta_{d-1}$, one *simply* has to define a random vector $\mathbf{\Gamma} \in [0,+\infty)^d$ with unit expectation.

**Definition 2.5.** Let $F$ be a cdf satisfying the assumptions of Theorem 2.4. The stdf $\ell_F$ is said to be *discrete*, with $N$ atoms, when its D-norm is related to a generator $\mathbf{\Gamma}$ whose distribution is *discrete* and made of $N$ atoms.

**Example 2.6.** *Let us consider the special case where $d = N$. Note that it can be typically the case of the input of a generative model with a latent distribution of dimension $N$. Let $F$ be a cdf in $\mathbb{R}^N$ with independent margins. Then, let*

$$\ell_F(\boldsymbol{x}) = \sum_{i\in[N]} x_i, \qquad \forall \boldsymbol{x} \in (0,\infty)^N.$$

*This stdf is discrete with $N$ atoms: indeed, one can easily check that, if $\mathbf{\Gamma}$ has the discrete uniform distribution on $\{N\,\boldsymbol{e}_1, \ldots, N\,\boldsymbol{e}_N\}$, then $\mathbb{E}\,[\mathbf{\Gamma}_i] = 1$ for any $i \in [N]$, and its D-norm is given by*

$$\|\boldsymbol{x}\|_D = \mathbb{E}\left[\max_{i\in[N]}(x_i \mathbf{\Gamma}_i)\right] = \frac{1}{N}\sum_{i\in[N]} x_i\, N = \ell_F(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in (0,\infty)^N.$$

*With similar arguments, it is easily checked that the spectral measure is the discrete uniform distribution on $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_N\}$.*

The concept of Definition 2.5 is at the core of our numerical scheme, since we will consider approximations based on discrete distributions for the generator $\mathbf{\Gamma}$ in Section 2.4.

**Characterization of simple max-stable distribution using D-norms and homogeneous function.** We aim at giving another criterion to establish the max-domain of attraction of a simple max-stable distribution. It will be crucially used in the proof of our main result (Theorem 2.12).

**Definition 2.7** (1-homogeneous function)**.** A function $h : [0,\infty)^d \to [0,\infty)^{d'}$ is 1-homogeneous if it satisfies:

$$\forall \boldsymbol{x} \in [0,\infty)^d, \;\; \forall \lambda \geq 0 : \;\; h(\lambda \boldsymbol{x}) = \lambda h(\boldsymbol{x}).$$

The set of 1-homogeneous continuous functions from $\mathcal{X}$ to $\mathcal{Y}$ is denoted by $\mathbb{H}_1(\mathcal{X}, \mathcal{Y})$.

The next result claims that identifying the stdf (or equivalently the D-norm) of a cdf $F$ (with asymptotically simple max-stable distribution) is equivalent to analyze tails of 1-homogeneous transforms of the associated random vector.

**Theorem 2.8** ([FF21, Theorem 4.4]). *Let $\boldsymbol{X}$ be a random vector in $[0,+\infty)^d$ with continuous cdf $F$ and let $\|.\|_D$ be a D-norm in $\mathbb{R}^d$ generated by $\boldsymbol{\Gamma}$. The following equivalence holds:*

$$\forall \boldsymbol{x} \in (0,\infty)^d : \quad F^t(t\boldsymbol{x}) \underset{t\to\infty}{\longrightarrow} \exp\left\{-\left\|\frac{1}{\boldsymbol{x}}\right\|_D\right\} \tag{2.4}$$

$$\iff \qquad \forall h \in \mathbb{H}_1([0,\infty)^d,[0,\infty)) : \quad t\mathbb{P}\left(h(\boldsymbol{X})>t\right) \underset{t\to\infty}{\longrightarrow} \mathbb{E}\left[h(\boldsymbol{\Gamma})\right]. \tag{2.5}$$

An indirect consequence of the above theorem is that the limiting value $\mathbb{E}\left[h(\boldsymbol{\Gamma})\right]$ does not depend on the chosen generator of the D-norm, when $h$ is a continuous 1-homogeneous function, see [FF21, Theorem 3.5] for a direct proof.

## 2.3 Generative Adversarial Networks

**Neural networks.** We refer to [Mur22, Chapter 13] for a thorough introduction on neural networks. Let us consider fully connected neural networks, also called Multi Layer Perceptron (MLP). These neural networks consist of a chain of $L$ operations of the form:

$$\boldsymbol{z}_l = f_l\left(\boldsymbol{z}_{l-1}\right) = \varphi_l\left(\boldsymbol{b}_l + \mathbf{W}_l \boldsymbol{z}_{l-1}\right) \tag{2.6}$$

for $l \in [L]$, where $\boldsymbol{z}_l \in \mathbb{R}^{q_l}$, $\varphi_l : \mathbb{R}^{q_l} \to (\mathbb{R}^+)^{q_l}$ is an activation function, $q_l$ is the number of neurons on the layer $l$, $\boldsymbol{W}_l$ is a matrix and $\boldsymbol{b}_l$ is a bias vector. Note that $\boldsymbol{z}_0$ corresponds to the $q_0$-dimensional input of the network, while $\boldsymbol{z}_L$ is the $q_L$-dimensional output. In this work, we focus on neural networks with ReLU activation functions: $\varphi_l(\boldsymbol{z}) = \max\{\boldsymbol{z}, \boldsymbol{0}\}$. The parameters of the network are:

$$\theta = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L, \boldsymbol{W}_1, \ldots, \boldsymbol{W}_L\} \in \Theta \tag{2.7}$$

(activation functions are fixed), where $\Theta$ is the space of parameters, the full Euclidean space without constraints.

**Proposition 2.9.** *A neural network parameterized by (2.6) with ReLU activation functions is a piecewise affine function. It is a continuous 1-homogeneous function when the bias vectors are zero.*

The proof is easy and left to the reader. Combining Proposition 2.9 with Theorem 2.8 shows that a distribution (with simple max-stable distribution) transformed by a ReLU neural network remains a heavy-tailed distribution with the same tail parameter $\gamma$.

**Generative Adversarial Networks.** Generative modeling is the task of approximating at best a distribution of interest, here written $\boldsymbol{X} \sim p_{\text{data}} \in \mathbb{R}^d$, and being able to draw samples from it. One way to do is to find a map $\mathcal{G}$ which takes as input a random variable $\boldsymbol{Z} \sim p_{\boldsymbol{Z}} \in \mathbb{R}^N$ which is easy to sample, such that:

$$\mathcal{G}(\boldsymbol{Z}) \overset{d}{\approx} \boldsymbol{X}.$$

The random vector $\boldsymbol{Z}$ is commonly referred as the latent noise and $p_{\boldsymbol{Z}}$ the latent distribution. Generative Adversarial Networks [GPAM+14] are widespread generative models. GANs

consist of a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$. GANs play an adversial game where the generator tries to mimic the true distribution generating the data whereas the discriminator tries to uncover fake data points simulated by the generator. Namely, GANs optimize for:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{D}, \mathcal{G}) \quad \text{where} \quad V(\mathcal{D}, \mathcal{G}) = \mathbb{E}_{\boldsymbol{X} \sim p_{\text{data}}} \left[\log \mathcal{D}(\boldsymbol{X})\right] + \mathbb{E}_{\boldsymbol{Z} \sim p_{\boldsymbol{Z}}} \left[\log(1 - \mathcal{D}(\mathcal{G}(\boldsymbol{Z})))\right], \tag{2.8}$$

where $p_{\text{data}}$ is the distribution of the data and $p_{\boldsymbol{Z}}$ is the latent distribution. Usually, $p_{\text{data}}$ is chosen among the uniform or the Gaussian distribution. Both $\mathcal{G}$ and $\mathcal{D}$ are neural networks. In this work, we consider neural networks parameterized with MLPs (2.6), and, with the notations of (2.6), note that $q_0 = N$ and that $q_L = d$.

## 2.4 New approximation results for the stdf

Our goal is to assess how well the dependence in the extremes of a target distribution can be by approximated with a GAN based on heavy-tailed independent input noise. To this end, our first result establishes the nature of the stdf of a neural network output, when used with heavy-tailed input noise.

**Proposition 2.10.** *Consider a random vector $\boldsymbol{Z}$ with a discrete stdf $\ell_{\boldsymbol{Z}}$ with $N$ atoms. Consider a neural network $G_{\theta,N}$, with ReLU activation functions and $L-1$ hidden layers, where $\theta$ is the parametrization of the network (2.7) and $N$ is the dimension of the input noise ($q_0 = N$). Assume that all the matrix weights $(\boldsymbol{W}_1, \cdots, \boldsymbol{W}_L)$ are non-negative. Then, the stdf of the output $G_{\theta,N}(\boldsymbol{Z})$ is discrete with at most $N$ atoms.*

See Section B.1 for the proof. It appears that the output of a neural network with heavy-tailed input noise has a discrete stdf. The next result states that it is possible to attain perfect reproduction of any discrete stdf with MLP neural networks.

**Proposition 2.11.** *Consider a heavy-tailed random vector $\boldsymbol{X}$ with a discrete stdf $\ell$ with $N$ atoms. There exists a neural network $\mathcal{G}_{\theta^*,N}$ with parametrization $\theta^* \in \Theta$ and input noise $\boldsymbol{Z} \in \mathbb{R}^N$ with i.i.d. unit Fréchet margins such that:*

$$\ell = \ell_{\mathcal{G}_{\theta^*,N}(\boldsymbol{Z})}$$

*where $\ell_{\mathcal{G}_{\theta^*,N}(\boldsymbol{Z})}$ is the stdf of $\mathcal{G}_{\theta^*,N}(\boldsymbol{Z})$.*

In words, any discrete stdf can be perfectly approximated by a neural network with i.i.d. unit Fréchet margins as input noise with sufficiently large dimension.
When the stdf of interest is not discrete, our final result proves that it can be nevertheless approximated using a MLP neural network with an arbitrary precision.

**Theorem 2.12.** *Let $\boldsymbol{X}$ be a random vector in $\mathbb{R}^d$, with continuous cdf $F$, belonging to the max-domain of some distribution. Denote by $\ell_F$ its stdf that is related to a $D$-norm (Theorem 2.4). Consider using a MLP neural network $\mathcal{G}_{\theta,N}$ with a single layer (i.e. $L = 1$) and with as latent noise a random vector of $N$ i.i.d. unit Fréchet margins, and denote by $\ell_{\mathcal{G}_{\theta,N}}$ its stdf. Then,*

$$\inf_{\theta \in \Theta} \sup_{\boldsymbol{x} \in (0,\infty)^d} \frac{\left|\ell_F(\boldsymbol{x}) - \ell_{\mathcal{G}_{\theta,N}}(\boldsymbol{x})\right|}{\|\boldsymbol{x}\|_\infty} \leq \epsilon(N), \tag{2.9}$$

*where*

$$\epsilon(N) \sim C(d) N^{-1/(d-1)} \text{ as } N \to \infty, \tag{2.10}$$

*where $C(d)$ is a constant depending on $d$ and $\ell_F(\mathbf{1})$. The bound (2.9) is achieved with one layer neural networks.*

In this previous theorem, note that the error bound is a worst case scenario. For example, when the stdf is discrete, the error is zero for $N$ sufficiently large in virtue of Proposition 2.11. Let us highlight that, in view of the upper bound (2.10), the attainable precision of the neural network approximation increases with the dimension of the latent noise. However, the higher the dimension $d$, the slower is the convergence. One can then expect that accurate approximations of the dependence in high dimension would require high dimensional latent noise, which is consistent with the intuition. This will be illustrated in the Numerical Experiments section.

# 3 Numerical experiments

First, we describe in Section 3.1 the proposed algorithms for training and sampling from a dataset presenting heavy-tailed characteristics. Second, experiments performed on synthetic datasets generated using the Gumbel copula are detailed in Section 3.2 to compare the proposed method with the standard GAN approach. Finally, experiments on the stock market index S&P500 are presented in Section 3.3.

## 3.1 Algorithms and implementation

### 3.1.1 Algorithms

We accommodate the original GAN algorithm [GPAM+14] to have a heavy-tailed latent noise. This idea originates from [HCL+21]. Algorithm 1 describes the learning phase while Algorithm 2 summarizes the generation phase.

### 3.1.2 Implementation

**Code.** The implementation is based on a publicly available GitHub repository[1] using the machine learning library PyTorch[2]. We adapt the source code so that both the generator and discriminator are fully connected MLPs (2.6). For writing a specific architecture of a neural network, we use a list that details all hidden layers, that is layers $q_l$ for $l \in [L-1]$ in (2.6). As an example, a network parametrized by $[100, 200]$ means that it has 2 hidden layers, $L = 3$, $q_1 = 100$ and $q_2 = 200$. For the activation functions, we use LeakyReLu [MHN13]: note that our theoretical result (Theorem 2.12) still holds since a LeakyReLu is a difference of two ReLUs. For gradient descent, we use the optimizer Adam [KB14] and we do not tune the base parameters. Early stopping is used for regularization.
In these experiments, we allow the networks to have several hidden layers. The proof of Proposition 2.11 only involves networks of depth one to handle the dependence in the extremes, but higher depths may be necessary to reproduce the dependence in the tails or in the bulk of the distribution. In (B.27), it is seen that a one layer transformation produces a heavy-tail term (left term) and a bounded term. Our aim is that on one side, we manage to catch the behaviour of the distribution tail with the left unbounded, 1-homogenous term and, on the other side, we hope to capture the behaviour of the bulk with the right bounded term.

**Hardware.** All experiments were performed on the cluster Cholesky[3] of Ecole Polytechnique. Code was run on Intel Xeon CPU Gold 6230 20 cores @ 2.1 Ghz with 192 GB of memory.

[1] https://github.com/eriklindernoren/PyTorch-GAN?tab=readme-ov-file#gan
[2] https://pytorch.org/
[3] https://docs.idcs.mesocentre.ip-paris.fr/cholesky/hardware_description/

**Algorithm 1:** Generative modelling: learning algorithm

**input :**

- Dataset $\left\{\boldsymbol{X}^{(1)}, \cdots, \boldsymbol{X}^{(n_{\text{data}})}\right\}$ of $n_{\text{data}}$ i.i.d. sample points in $\mathbb{R}^d$.
- Fully connected initialized generator $\mathcal{G}$ and discriminator $\mathcal{D}$.
- Estimators $\hat{F}_1, \ldots, \hat{F}_d$ of marginal cdf.
- Tail parameter $\gamma > 0$ for the renormalization step.
- GAN hyperparameters: Network architecture, optimization parameters and callbacks (early stopping, maximum number of iterations ...). The parameters are dependent on the chosen implementation. See Section 3.1.2 for more details.

**begin**

1. For any $j \in [d]$, estimate the marginal cdf of $X_j$ on the training set $\left\{X_j^{(1)}, \ldots, X_j^{(n_{\text{data}})}\right\}$ by

$$x \in \mathbb{R} \mapsto \hat{F}_j(x) \in [0, 1].$$

2. Transform the marginals to Fréchet distributions with tail parameter $\gamma$:

$$\forall i \in [n_{\text{data}}], j \in [d] : \tilde{X}_j^{(i)} = \left(\frac{1}{1 - \hat{F}_j\left(X_j^{(i)}\right)}\right)^{\gamma}. \tag{3.11}$$

3. Train GAN over transformed data $\left\{\tilde{\boldsymbol{X}}^{(1)}, \ldots, \tilde{\boldsymbol{X}}^{(n_{\text{data}})}\right\}$: optimize (2.8) where $p_{\text{data}}$ is a componentwise Pareto distribution with tail parameter $\gamma$ and independent margins.

**return** *the optimal generator $\mathcal{G}^\star$, and the estimators $\hat{F}_1, \ldots, \hat{F}_d$ of the margins.*

**end**

**Algorithm 2:** Generative modelling: sampling algorithm

**input :**

- Number of data points to sample $n_{\text{sample}}$.
- Tail parameter $\gamma > 0$.
- Estimators $\hat{F}_1, \ldots, \hat{F}_d$ of the margins.
- Generator $\mathcal{G}^\star$.

**begin**

1. Sample $n_{\text{sample}}$ points $\left(\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(\text{nsample})}\right)$ from $p_{\text{data}}$, a componentwise Pareto distribution with tail parameter $\gamma$ and independent margins.
2. Transform the noise through the generator:

$$\forall i \in [n_{\text{sample}}] : \ \tilde{\boldsymbol{X}}^{(i)}_{\text{sample}} = \mathcal{G}^\star(\boldsymbol{z}^{(i)}).$$

3. Transform the sample points $\tilde{\boldsymbol{X}}_{\text{sample}}$ to the original scale with the inverse of the estimated margins:

$$\forall i \in [n_{\text{sample}}], j \in [d] : \ \hat{X}_j^{(i)} = \hat{F}_j^{-1}(\tilde{X}_j^{(i)}). \tag{3.12}$$

**end**

**return** *Sampled points* $\left\{\hat{\boldsymbol{X}}^{(1)}, \ldots, \hat{\boldsymbol{X}}^{\left(n_{\text{sample}}\right)}\right\}$.

## 3.2 Simulated data from a Gumbel copula with Pareto margins

We first propose in Section 3.2.1 a visual illustration of the behaviour of heavy-tailed latent noise models on a two dimensional dataset. Second, we investigate in Section 3.2.2 the performance of Algorithm 1 on higher dimensional data and perform a comparison with the baseline GAN algorithm using Gaussian input noise. Specifically, we study the model's ability to reproduce data with both given dependence degree and tail heaviness. To this end, the target distribution is obtained by combining a Gumbel copula with Pareto margins. We refer to Section A.4 in the Appendix for basic material on copulas.

### 3.2.1 Two-dimensional data: an illustration

Let us first consider a toy dataset of size $n_{\text{data}} = 10,000$ generated by a Gumbel copula in dimension $d = 2$ with identically distributed Pareto margins, see the left panel of Figure 1 for an illustration. It appears that, in the tail, data are evenly distributed along the horizontal axis $\boldsymbol{e}_1$ and the vertical one $\boldsymbol{e}_2$. As a comparison, the right panel of Figure 1 displays a plot of the input noise with independent Pareto random margins. In contrast to the Gumbel training data, the independent Pareto noise whence conditioned to being large in norm, is concentrated in the two directions $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. This is due to the fact that the spectral measure associated with independent Pareto random variable is discrete, with weights on $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. The left panel of Figure 2 displays the simulated data obtained with a GAN trained on the previously described dataset using $N \in \{2, 3, 5, 10\}$ independent Pareto random variables as input noise. The right panel displays the histogram of the angles defined by $\arccos(\hat{X}^{(1)}/\|\hat{\boldsymbol{X}}\|)$ associated with 10% largest Euclidean norms $\|\hat{\boldsymbol{X}}\|$. For small latent dimensions $N \in \{2, 3\}$, one can notice spikes in the distribution tail: Generated datasets exhibit as many spikes as the dimension of the latent noise. Indeed, the spectral measure of i.i.d. Pareto noise is discrete, and, in accordance with Proposition 2.10, transforming a $N$-dimensional independent Pareto noise (having a discrete spectral measure with $N$ atoms) through a 1-homogeneous mapping yields a distribution with a discrete spectral measure involving at most $N$ atoms. For larger dimensions $N \in \{5, 10\}$, spikes become less visible but it is still possible to distinguish them. In these higher dimensions, it visually seems that the approximation of the dependence gets better.

### 3.2.2 Higher dimensional experiments

After a thorough understanding of the behaviour of the model in a two dimensional setting, we explore its performance on higher dimensional datasets, still using the Gumbel copula, but with a variety of dependence and tail coefficients.

**Experimental setup.** The trained data are sampled from the $d$-dimensional Gumbel copula (see Section A.4 in the Appendix) with $d \in \{2, 5, 10, 20, 50\}$ and dependence parameter $\beta \in \{4/3, 2, 4\}$, leading to Kendall's $\tau \in \{1/4, 1/2, 3/4\}$. See Equation (A.25) for a definition of Kendall's $\tau$. The margins are chosen to be Pareto distributed with shape parameter $\alpha \in \{1.5, 2.0, 2.5\}$. In each of the $5 \times 3 \times 3 = 45$ considered situations, $n_{\text{data}} = 10,000$ data points are simulated for training and $20,000$ data points are considered for testing. Note that the testing set is larger than the training set in order to assess the ability of GANs to extrapolate to extreme regions unseen in training. In this synthetic case, we provide the tail-index $\gamma = 1/\alpha$ to Algorithm 1 and focus on the quality of reproduction of dependence in extreme regions.

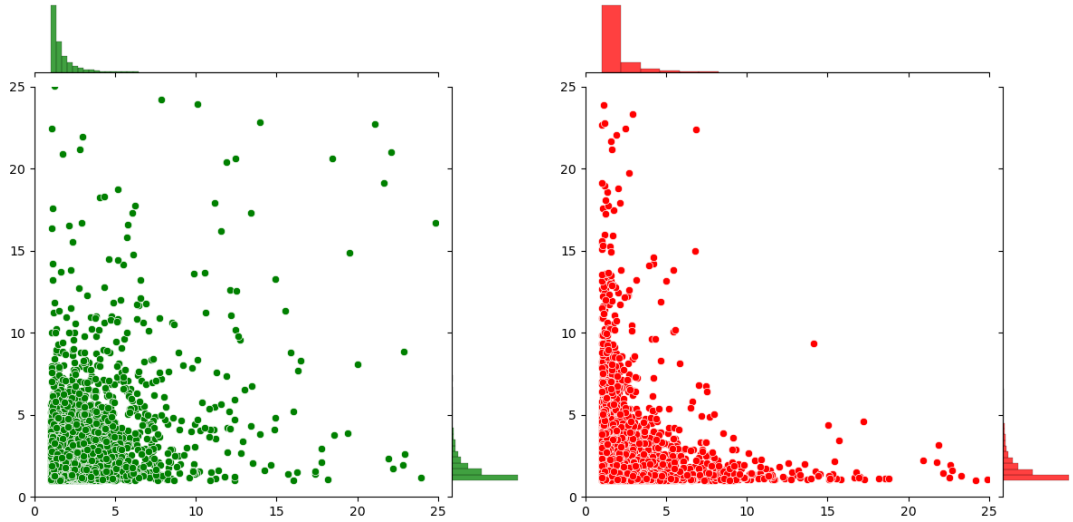


Figure 1: Illustration in a two-dimensional setting on simulated data of size $n_{\text{data}} = 10,000$ with Pareto margins and tail parameter $\alpha = 2.0$. Left panel: sample from a Gumbel copula with dependence parameter $\beta = 4/3$. Right panel: sample with two independent margins.

**Metrics.** Two metrics are considered for assessing the performance of the GANS: the Absolute Kendall Error (AKE), see [AGG22, Section 3.2 and Appendix A] for a definition and some background, and the Sliced Wasserstein Distance (SWD), see Equation (A.26) in the Appendix. Since the focus is on extreme regions, we compute the Euclidean norm of all data points and select data points whose norms are larger than the upper $\xi$-quantile, with $\xi \in \{90\%, 95\%, 99\%\}$. Both metrics are then computed (denoted by $\text{AKE}_\xi$ and $\text{SWD}_\xi$) on the resulting points. This operation is performed both on the sample generated by GAN methods and on the dataset simulated from the true distribution. These new subsamples are then normalized on the Euclidean sphere, where the Sliced Wasserstein distance is estimated.

**Preliminary hyperparameter search.** We explore a number of possible parametrizations of GAN models within a selected range of hyperparameters. For both the generator and the discriminator, fully connected neural networks are used with the following 15 hidden dimensions: $[50]$, $[75]$, $[100]$, $[125]$, $[200]$, $[300]$, $[400]$, $[100, 100]$, $[200, 200]$, $[300, 300]$, $[400, 400]$, $[100, 200, 100]$, $[200, 400, 200]$, $[100, 200, 200, 100]$ and $[200, 400, 400, 200]$. For the latent dimension, we explore values $N$ ranging from 1 to 200. For the optimization parameter, values ranging (log-uniformly) from $10^{-6}$ to $10^{-1}$ are tested. We investigate the use of four batch sizes: $\{128, 256, 512, 1024\}$. A Bayesian search (see [WCZ+19]) is implemented using weight and biases' sweep tool[4]. At first, some parametrizations are selected at random and, next, the probability for a configuration to be selected is updated with the use of a Bayesian rule. See the referred online documentation for more details on Bayesian hyperparemeter tuning. Considering the top performing models of this research, the following conclusions can be drawn on the choice of hyper parameters: Among tested models, best performing generators (80%+) have a light parametrization (fewer parameters) $[100]$ or $[200]$, while best performing discriminators (80%+) have a heavy parametrization (more parameters) $[100, 200, 200, 100]$ or

[4]https://docs.wandb.ai/guides/sweeps

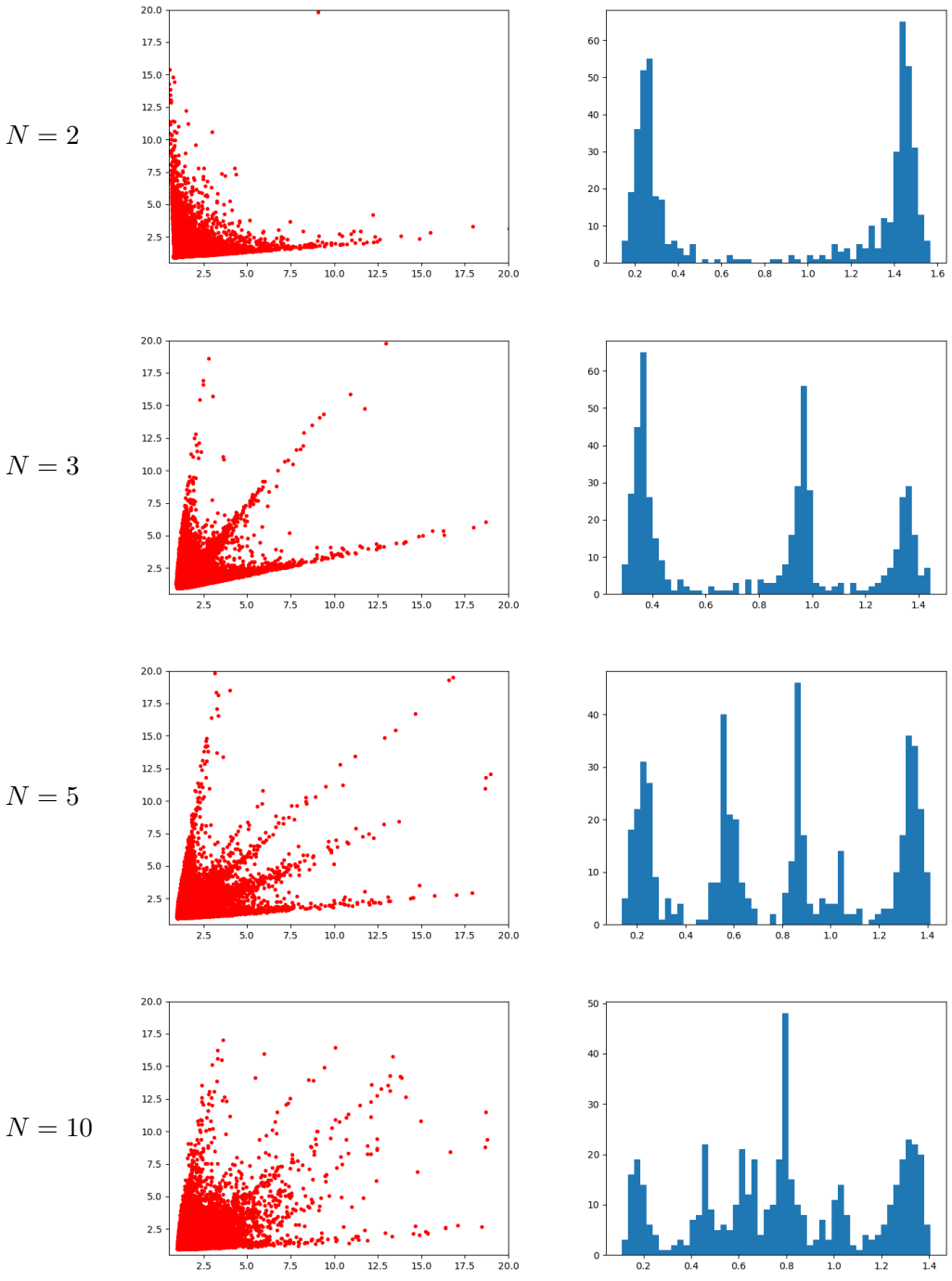


Figure 2: Illustration in a two-dimensional setting on simulated data of size $n_{\text{data}} = 10,000$ with Pareto margins and tail parameter $\alpha = 2.0$. Left panel: generated data using a GAN with independent Pareto margins of latent dimension $N \in \{2, 3, 5, 10\}$. Right panel: histogram of angles for data with 10% largest Euclidean norms.

$[200, 400, 400, 200]$. Bigger batch sizes yield better performances. Good values for the learning rate include range $\left[5 \cdot 10^{-6}, 5 \cdot 10^{-4}\right]$.

**Setting up models for comparison.** After this hyperparameter exploration, two versions of GAN are compared: the base version of GAN with a Gaussian noise $\boldsymbol{Z} \sim \mathcal{N}(0, \boldsymbol{I}_{N\times N})$, referred to as LTGAN (LT for light-tailed), and our version, Algorithm 1, referred to as HTGAN (HT for heavy-tailed). Thanks to the previous insights on the preliminary hyperparameter search, we are able to choose a subset of evaluation hyperparameters to compare both models. Both models are run for each configuration. In order to have an exhaustive comparison, a grid search is performed on the following range of hyperparameters: $N \in \{2, 5, 10, 20, 50, 80\}$, hidden dimensions ranging in $\{[100, 200, 100], [200, 400, 200], [100, 200, 200, 100], [200, 400, 400, 200]\}$ for the discriminator, in $\{[100], [200]\}$ for the generator, $\{10^{-4}, 10^{-5}\}$ for the learning rate.

**Results.** It appears in Table 1 that HTGAN performs better than LTGAN with respect to all six metrics and all considered dependence coefficients $\beta \in \{4/3, 2, 4\}$ in the heavier tail setting $\alpha = 1.5$ (all percentages are above 50%). For $\alpha = 2.5$, it appears that HTGAN fails to outperform the baseline on the $\beta = 4$, $\alpha = 2.5$ setting. It suggests that in low tail-index settings, light-tailed outputs still manage to provide a good approximation of the target distribution, even thought the target is heavy-tailed. For almost all experiment settings, HTGAN is comparatively better than LTGAN for the highest tail coefficient. For the sliced Wasserstein distance, it appears that HTGAN gets comparatively better as both $\beta$ and $\gamma = 1/\alpha$ increase. This result is coherent with our theoretical development: It is known that (piecewise) linear transformations of Gaussian variables cannot generate (asymptotic) dependence in the tails [WKKK20], a case which is met when $\beta$ gets larger and larger.
Figures 3 and 4 plot the performance of both methods for $\alpha = 1.5$ and $\alpha = 2.5$, with varying values of $\beta$. With $\alpha = 1.5$, the point cloud is well above the median and our method is better performing than the baseline. In the case $\alpha = 2.5$, the difference is not neat and no clear conclusion can be drawn.

Table 1: Percentage (%) of parametrizations for which HTGAN is better than a LTGAN for the six considered metrics. Results are given with precision $\pm 0.1\%$. Values are averaged over data dimensions $d \in \{2, 5, 10, 20, 50\}$.

| | $\beta = 4/3$ | | | $\beta = 2$ | | | $\beta = 4$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $\alpha$ | 1.5 | 2 | 2.5 | 1.5 | 2 | 2.5 | 1.5 | 2 | 2.5 |
| AKE_90 | 65.8 | 68.1 | 65.6 | 80.4 | 78.4 | 66.7 | 88.7 | 86.6 | 72.6 |
| AKE_95 | 65.6 | 69.5 | 66.4 | 79.7 | 79.4 | 69.2 | 87.8 | 87.2 | 75.1 |
| AKE_99 | 62.5 | 67.4 | 65.1 | 77.0 | 80.1 | 72.7 | 86.8 | 88.0 | 78.4 |
| SWD_90 | 81.7 | 69.1 | 45.9 | 76.7 | 67.7 | 52.6 | 63.7 | 54.7 | 41.4 |
| SWD_95 | 84.5 | 74.3 | 51.1 | 76.9 | 66.7 | 54.3 | 62.0 | 52.2 | 39.5 |
| SWD_99 | 86.4 | 81.5 | 64.8 | 77.5 | 59.5 | 49.6 | 56.9 | 48.3 | 34.4 |

**Latent dimension.** Equation (2.9) gives evidence that a higher value of the latent dimension $N$ is necessary to obtain a better approximation of the dependence structure. For each experiment specification (*i.e.* a value for $\alpha$ and for $d$), we have performed a Bayesian hyperparameter search [WCZ+19] with 400 runs, with the same hyperparameter ranges as for the

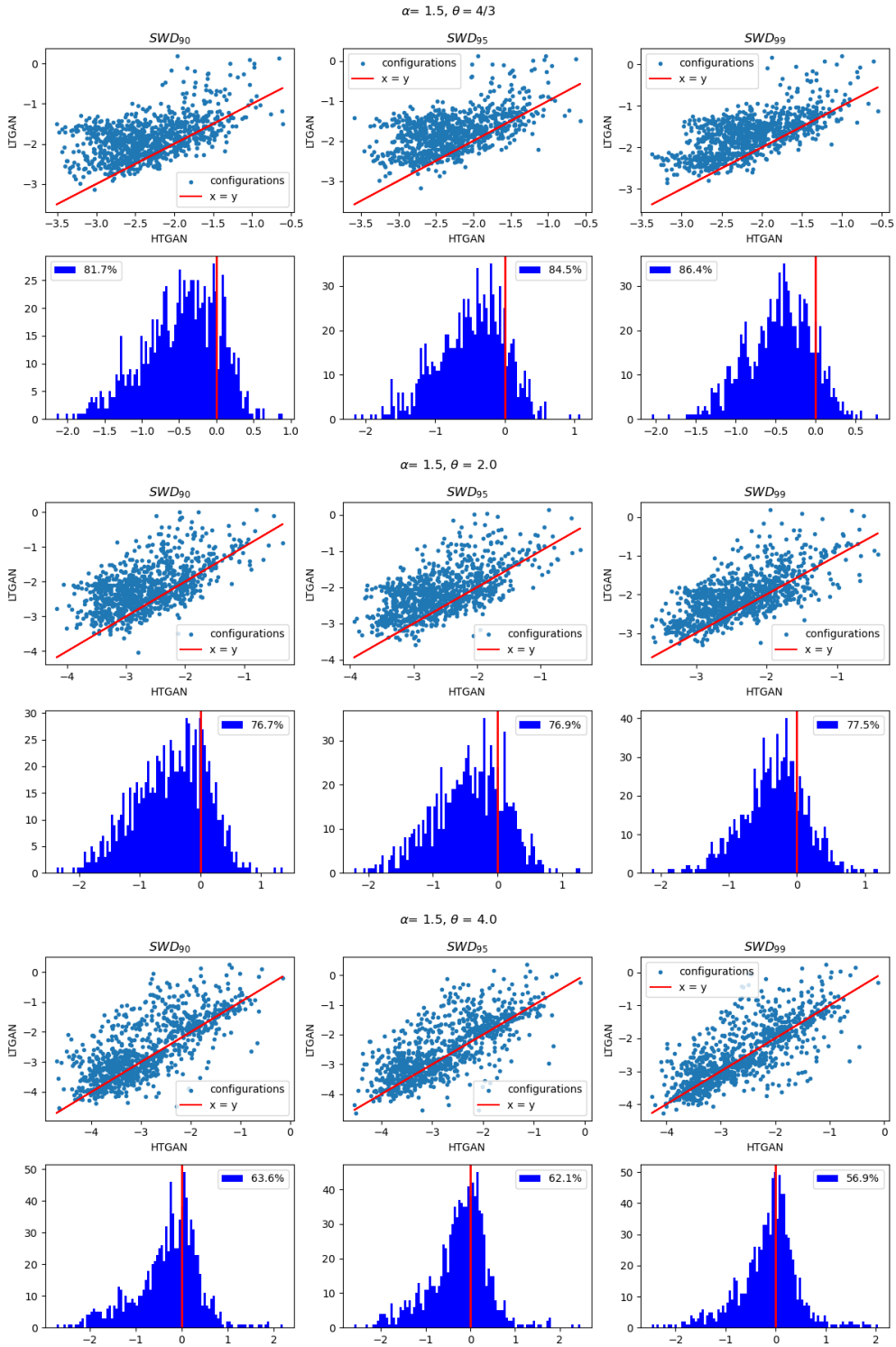


Figure 3: Illustration in a $d$-dimensional setting on simulated data of size $n_{\text{data}} = 10,000$ from a Gumbel copula (with dependence parameter $\beta$) and Pareto margins (with tail parameter $\alpha = 1.5$). Each three groups of $2 \times 3$ plots corresponds to one experimental setting, *i.e.* a specification of $\alpha$ and $\beta$. Figures are averaged over dimensions $d \in \{2, 5, 10, 20, 50\}$. In each of these three groups: the first row plots the specified metric of HTGAN vs LTGAN in log scale for a given hyperparameter configuration. The second row is a histogram of the difference of the log of the metric for HTGAN vs LHTGAN for each parametrization. The legend corresponds to the proportion of cases where HTGAN performs better.

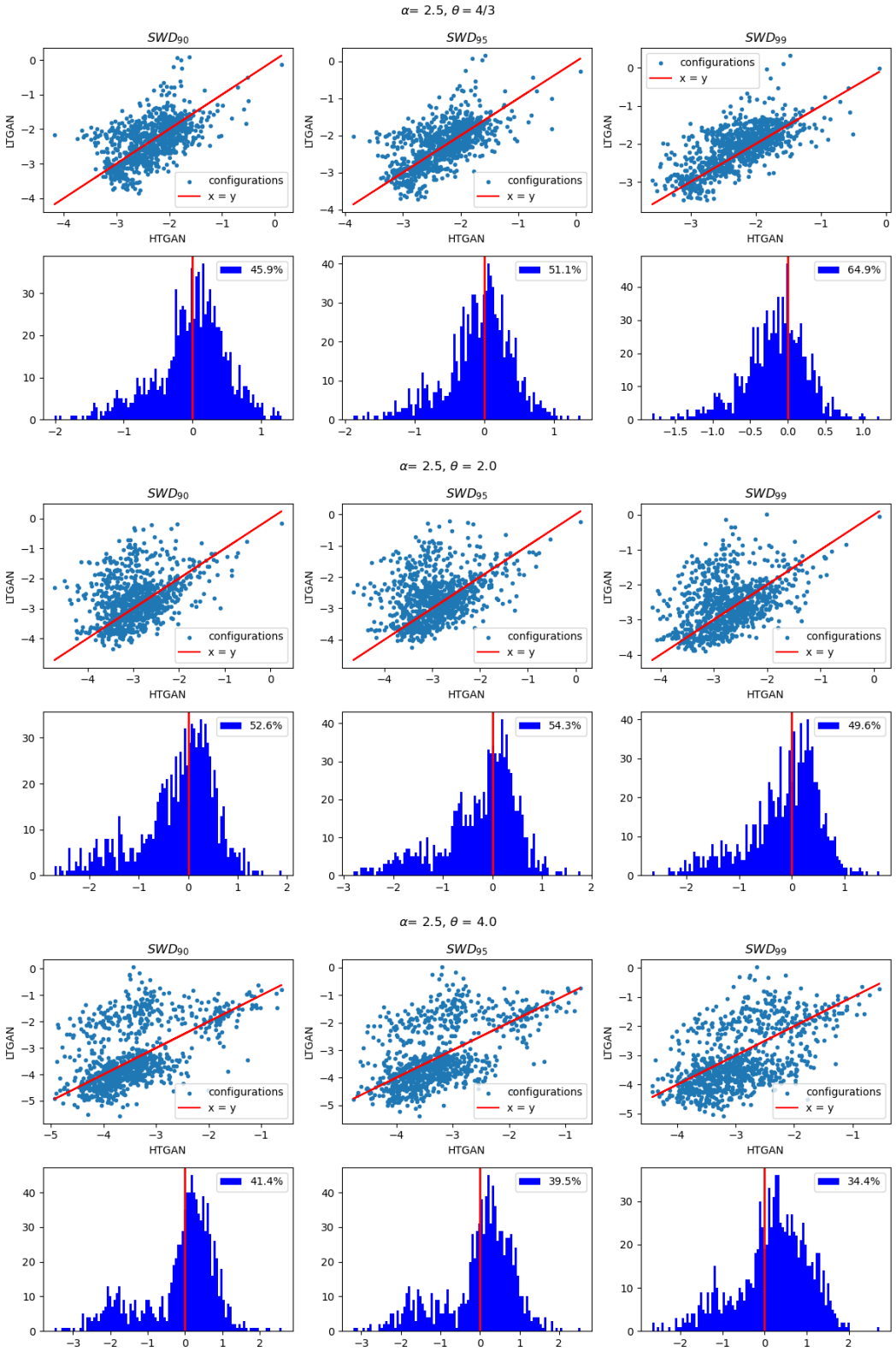


Figure 4: Illustration in a $d$-dimensional setting on simulated data of size $n_{\text{data}} = 10,000$ from a Gumbel copula (with dependence parameter $\beta$) and Pareto margins (with tail parameter $\alpha = 2.5$), $d \in \{2, 5, 10, 20, 50\}$. See Figure 3 for further details.

Table 2: Statistics (mean, min, max) on three on the top 5% best performing runs of HTGAN (best performing w.r.t. the SWD_90 metric). Statistics are averaged over $\beta \in \{4/3, 2, 4\}$.

| | | $N$ (latent dimension) | | | $\hat{\alpha}$ | | | SWD_90 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\alpha$ | $d$ | mean | min | max | mean | min | max | mean | min | max |
| 1.5 | 2 | 75.96 | 10 | 195 | 1.53 | 1.16 | 1.72 | 0.03 | 0.01 | 0.06 |
| | 5 | 91.07 | 17 | 197 | 1.61 | 1.20 | 1.96 | 0.04 | 0.01 | 0.06 |
| | 10 | 90.69 | 21 | 197 | 1.72 | 1.30 | 2.23 | 0.04 | 0.01 | 0.07 |
| | 20 | 91.33 | 26 | 182 | 1.77 | 1.23 | 2.64 | 0.03 | 0.01 | 0.07 |
| | 50 | 107.98 | 22 | 188 | 2.05 | 1.31 | 4.00 | 0.02 | 0.01 | 0.04 |
| 2.0 | 2 | 40.71 | 5 | 122 | 2.07 | 0.70 | 3.44 | 0.04 | 0.01 | 0.08 |
| | 5 | 55.96 | 11 | 164 | 2.41 | 1.63 | 3.68 | 0.04 | 0.01 | 0.12 |
| | 10 | 57.36 | 10 | 187 | 2.83 | 1.49 | 4.66 | 0.04 | 0.01 | 0.10 |
| | 20 | 64.60 | 10 | 191 | 3.03 | 1.06 | 4.88 | 0.03 | 0.01 | 0.07 |
| | 50 | 69.02 | 14 | 187 | 3.42 | 1.63 | 5.27 | 0.02 | 0.01 | 0.04 |
| 2.5 | 2 | 39.24 | 2 | 154 | 2.82 | 1.27 | 4.92 | 0.04 | 0.01 | 0.10 |
| | 5 | 50.20 | 7 | 177 | 3.54 | 1.33 | 7.63 | 0.05 | 0.01 | 0.13 |
| | 10 | 57.40 | 9 | 174 | 4.17 | 1.31 | 7.40 | 0.04 | 0.01 | 0.09 |
| | 20 | 72.98 | 12 | 175 | 4.32 | 0.97 | 8.79 | 0.03 | 0.01 | 0.06 |
| | 50 | 58.02 | 9 | 169 | 4.44 | 1.07 | 7.97 | 0.02 | 0.01 | 0.04 |

previous experiment section. Table 2 summarizes statistics on the latent dimension, $\hat{\alpha}$ and SWD_90 for the top 5% run on each experiment. It appears that, empirically, the theoretical result is verified: generally, for the various values of $\alpha$, as the dimension of the data increases, the optimal value of the latent dimension increases. We also note that, for each value of $\alpha$, as the dimension of the data increases, the estimate $\hat{\alpha}$ increases accordingly, which means lower tails. However, this conclusion is to be interpreted with caution in view of the great variability associated with the statistics (reflected by the values taken by min and max). This may be a consequence of the instability in training generative models with heavy-tailed inputs [HP24].

### 3.3 Experiments on real data

The proposed HTGAN method is tested on the publicly available dataset[5] consisting of daily data for companies of the S&P500. The S&P500 is a stock market index tracking of the 500 most valuable companies in the United States of America. The signal of interest is the absolute value of the normalized marginal returns for each stock:

$$r_t = \left| \left( X_{t+1} - X_t \right) / X_t \right| ,$$

which are bounded below by zero and only have an upper tail. The dataset consists of $n_{\text{data}} = 1259$ normalized marginal returns for each ticker. Companies are organized using the Global Industry Classification Standard (GICS). This classification divides companies into 11 sectors: Energy, Materials, Industrials, Consumer Discretionary, Consumer Staples, Health Care, Health Care, Financials, Information Technology, Communication Services, Utilities and Real Estate. Each sector is divided further into industry groups, industry and sub-industry. The returns of the S&P500 present a particular correlation structure illustrated in Figure 5 by a heatmap of the table consisting of estimated Kendall's tau for every pair of stock ticker,

[5] https://www.kaggle.com/datasets/camnugent/sandp500

see the Appendix for more details on Kendall's tau coefficient. It appears that returns are more correlated within each sector, especially in both the Financials and the Utilities sectors, than between sectors. Therefore, in the following, the performance of LTGAN and HTGAN are assessed on the two subsets associated with sectors Financials and Utilities, having both strong correlations (see Figure 5). Note that the number of tickers for the Financials and Utilities sector are not the same: $d = 27$ for the Financial sector, and $d = 57$ for the Utilities sectors.

**Sectors and subsectors.** The subsectors for the financial sector are: Asset Management & Custody Banks, Consumer Finance, Diversified Banks, Financial Exchanges & Data, Insurance Brokers, Investment Banking & Brokerage, Life & Health Insurance, Multi-Sector Holdings, Multi-line Insurance, Casualty Insurance, Regional Banks and Processing Services. For the Utilities sector, Subsectors are: Electric Utilities, Gas Utilities, Independent Power Producers & Energy Traders, Multi-Utilities and Water Utilities.

**Data normalization.** In the normalizing step (3.11) of Algorithm 1, two values of $\gamma = 1/\alpha$ are investigated: $\alpha = 1.5$ and $\alpha = 2.5$. The cumulative distribution function is estimated using its empirical counterpart:

$$\widehat{F}_j(x) = \frac{1}{n_{\text{data}} + 1} \sum_{i=1}^{n_{\text{data}}} \mathbb{I}\left\{X_j^{(i)} \leq x\right\}.$$

This transformation is not invertible, and therefore it is not possible to proceed to step (3.12) in the generation step of Algorithm 2. Therefore, we skip this step in the generation process. In our evaluation, we solely focus on the good reproduction of dependence in extreme regions (i.e. the difficult task). Our criteria do not assess how well the marginals are fitted.

**Results.** Figure 6 presents the results for experiments run on the S&P500 dataset. The results argue in favor of better performance of the heavy-tailed noise setting when data is renormalized to Pareto margins with a larger tail index (case $\alpha = 1.5$). This can be seen in the proportion of cases in which HTGAN performs better than LTGAN: 62.5%, 64.3% and 63.7% respectively for SWD_50, SWD_80 and SWD_90 metrics. In the case of a lighter tail, $\alpha = 2.5$, the performance of HTGAN is degraded, with relative performances of 38.7%, 30.4% and 34.5% respectively for SWD_50, SWD_80 and SWD_90 metrics. Therefore, for better accuracy in extreme regions, we argue in favor of a smaller $\alpha$ in the renormalization step.

# 4 Conclusion

We have provided a theoretical framework to analyse how a HTGAN behaves in extreme regions. We have established a bound on the quality of approximation of the stdf of the target distribution. For this theoretical development, we made an extensive use of the deep connections between Extreme Value Theory and the theory of D-norms. The numerical experiments support the theoretical findings. Especially, they demonstrate the superiority of the proposed method, HTGAN, in reproducing dependence in extreme regions compared to a standard LTGAN, on both a variety of synthetic heavy-tailed distribution exhibiting different dependence behaviours and a real dataset. We have also demonstrated the influence of the latent dimension in reproducing dependence in these regions. However, we have not addressed the

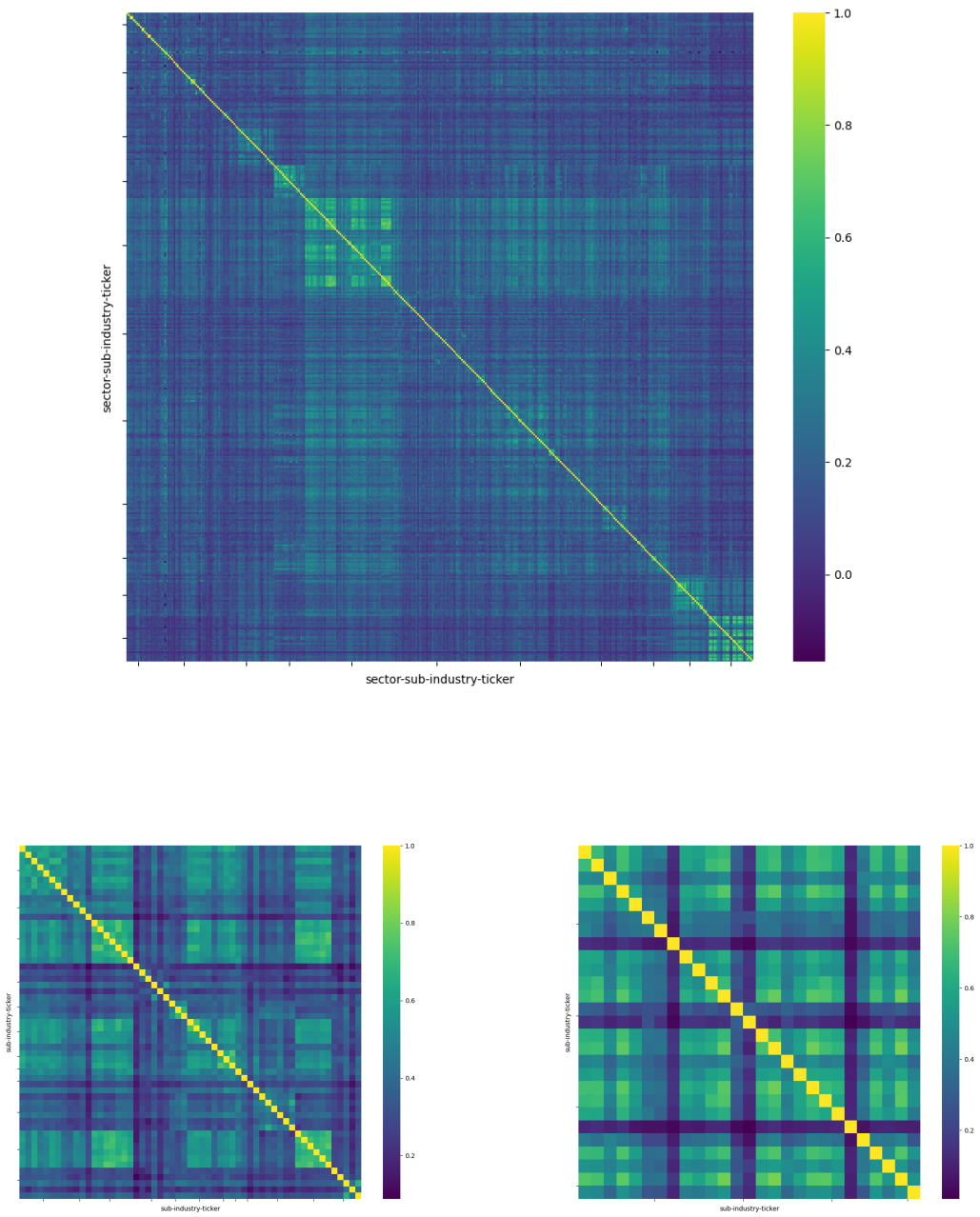


Figure 5: Estimated Kendall's $\tau$ for every pair of index considered. On top: every S&P500 ticker. Axis labeling corresponds to sectors of activity. On the bottom left: Financials sector. Right: Utilities sector. Labels for both correspond to subsectors of activity. A comprehensive list of sectors and subsectors is given in Paragraph 3.3.

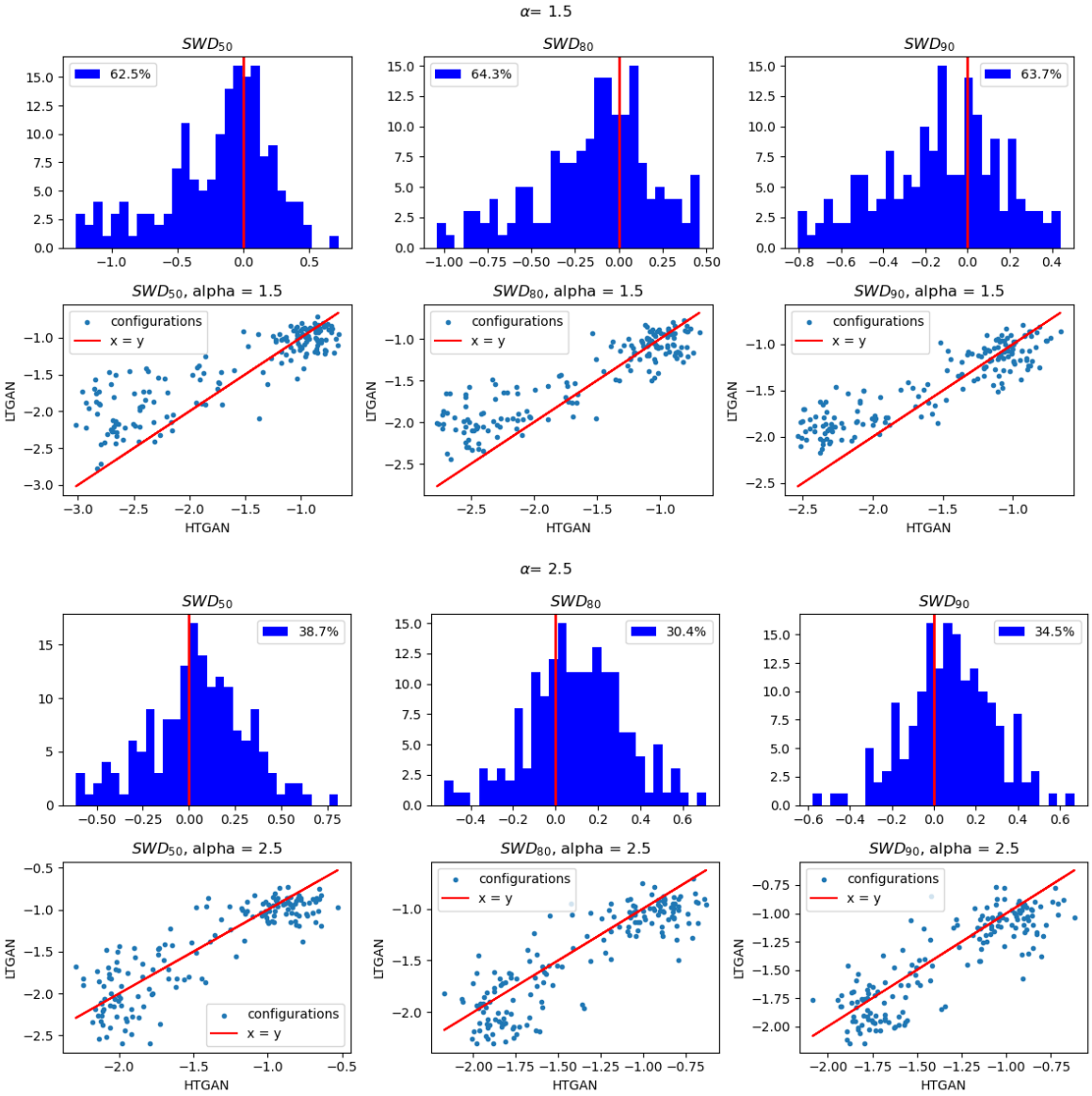


Figure 6: Results of the performance of HTGAN vs LTGAN on the Utilities ($d = 27$) and Financials subsectors ($d = 57$) of the S&P500 dataset. Results are aggregated with respect to the dimension $d$ for plot. Each two groups of $2 \times 3$ plots corresponds to one experimental setting, *i.e.* $\alpha = 1.5$ (top) and $\alpha = 2.5$ (bottom). In each of these two groups: the first row plots the specified metric of HTGAN vs LTGAN in log scale for a given hyperparametrization configuration. The second row is a histogram of the difference of the log of the metric for HTGAN vs the metric for LTGAN for each parametrization. The legend corresponds to the proportion of cases where HTGAN performs better.

issue of instability of training models with heavy-tailed inputs. This is left to further works. There also remains an ambiguity regarding a sensible choice for the tail parameter $\alpha$ in the marginal normalization step of the learning algorithm. We suggest that $\alpha$ should be treated as an hyperparameter of the method that is left to tune by the practitioner in the learning phase.

**Acknowledgments.**

The three authors acknowledge the support of the Chair "Stress Test, Risk Management and Financial Steering", led by the French Ecole Polytechnique and its Foundation and sponsored by BNP Paribas. S. Girard also gratefully acknowledges support from the French National Research Agency under the grant ANR-23-CE40-0009.

# A Theoretical background

Section A.1 provides additional background on extreme-value theory to complement Section 2.1. Section A.2 is devoted to D-norms, introduced in Section 2.2. Section A.3 focuses on the theory of quantization. Finally, Section A.4 provides some details on copulas and measures of dependence.

## A.1 Extreme-value theory

### A.1.1 General set-up

Consider a $\mathbb{R}^d$ valued random vector $\boldsymbol{X}$ with associated cdf $\boldsymbol{x} \in \mathbb{R}^d \mapsto F(\boldsymbol{x}) = \mathbb{P}(\boldsymbol{X} \leq \boldsymbol{x})$. The primary goal of extreme-value theory is to establish the asymptotic distribution of well normalized maxima of realizations of $\boldsymbol{X}$, see [Res87, BGTS04, HF06] for reference textbooks. Let $(\boldsymbol{X}_i)_{i\in[n]}$ be an i.i.d. sample from $F$ and consider

$$\boldsymbol{M}_n := \frac{\max_{i\in[n]} \boldsymbol{X}_i - \boldsymbol{b}_n}{\boldsymbol{a}_n},$$

where $\boldsymbol{a}_n > \boldsymbol{0}$ and $\boldsymbol{b}_n$ are normalizing sequences in $\mathbb{R}^d$. Clearly, the cdf of $\boldsymbol{M}_n$ is given by

$$F_{\boldsymbol{M}_n}(\boldsymbol{x}) = F^n(\boldsymbol{a}_n\boldsymbol{x} + \boldsymbol{b}_n).$$

This remark gives rise to the following definition:

**Definition A.1** (Max-Domain of Attraction (MDA), [HF06, Section 6.1.2])**.** A cdf $F$ on $\mathbb{R}^d$ is said to be in the max-domain of attraction of a cdf $G$, denoted by $F \in \mathrm{Dom}(G)$, if there exist sequences $\boldsymbol{a}_n > 0$ and $\boldsymbol{b}_n$, $n \in \mathbb{N}$ such that:

$$\forall \boldsymbol{x} \in \mathbb{R}^d : \; F^n(\boldsymbol{a}_n\boldsymbol{x} + \boldsymbol{b}_n) \underset{n\to\infty}{\longrightarrow} G(\boldsymbol{x}). \tag{A.13}$$

In other words, the normalized maximum of a sample from $F$ converges in distribution to a random vector with cdf $G$. It can be shown that the limiting distribution function is necessarily max-stable, as defined below:

**Definition A.2** (Max-stable distribution, [HF06, Section 6.1.2])**.** A cdf $G$ on $\mathbb{R}^d$ is called max-stable if, for any $n \in \mathbb{N}$, there exist sequences $\boldsymbol{a}_n > 0$ and $\boldsymbol{b}_n \in \mathbb{R}^d$ such that

$$\forall \boldsymbol{x} \in \mathbb{R}^d : \; G^n(\boldsymbol{a}_n\boldsymbol{x} + \boldsymbol{b}_n) = G(\boldsymbol{x}).$$

Besides, convergence (A.13) can be extended to a continuous counterpart; one has $F \in \mathrm{Dom}(G)$ if and only if there exist two functions $\boldsymbol{a}_t > 0 \in \mathbb{R}^d$ and $\boldsymbol{b}_t \in \mathbb{R}^d$ such that:

$$\forall \boldsymbol{x} \in \mathbb{R}^d : \; F^t(\boldsymbol{a}_t\boldsymbol{x} + \boldsymbol{b}_t) \underset{t\to\infty}{\longrightarrow} G(\boldsymbol{x}). \tag{A.14}$$

### A.1.2 Margins

Let us highlight that, if $F$ fulfills (A.13) or (A.14), then its margins $F_j$ satisfy

$$\forall x \in \mathbb{R} : \; F_j^n(a_{j,n}x + b_{j,n}) \underset{n\to\infty}{\longrightarrow} G_j(x),$$

where $G_j$ denotes the $j$th margin of $G$, $j \in [d]$. The *univariate* extreme-value theorem (see for instance [FT28, Gne43]) then provides a parametric form for the cdf $G_j$s which generalizes the parametrizations of Fréchet ($\gamma > 0$), Gumbel ($\gamma = 0$) and reverse-Weibull ($\gamma < 0$):

**Theorem A.3** ([HF06, Theorem 1.1.3]). *The $\mathbb{R}$-valued cdf's satisfying (A.13) are of the form $G_\gamma(a\cdot+b)$ with $a>0$, $b\in\mathbb{R}$ and $\gamma\in\mathbb{R}$, where $G_\gamma$ is a Generalized Extreme Value Distribution (GEVD) defined as*

$$\forall x\in\mathbb{R},\ 1+\gamma x>0:\ G_\gamma(x)=\exp\left\{-(1+\gamma x)^{-1/\gamma}\right\}. \tag{A.15}$$

*When $\gamma=0$, $G_0$ is interpreted as the pointwise limit in the above formula: $G_0(x)=\exp\{-e^{-x}\}$.*

Therefore, each margin $G_j$ of $G$ can be written as $G_j=G_{\gamma_j}$ following (A.15) with $\gamma_j\in\mathbb{R}$, $j\in[d]$. Let us note that, for each max-stable cdf $G$, one can consider $G_\star$, the so-called associated simple max-stable distribution, which is defined as:

$$\forall \boldsymbol{x}\in\mathbb{R}^d:\ G_\star(\boldsymbol{x})=G\left(\frac{\boldsymbol{x}^{\boldsymbol{\gamma}}-\mathbf{1}}{\boldsymbol{\gamma}}\right), \tag{A.16}$$

with $\boldsymbol{\gamma}=(\gamma_j)_{j\in[d]}\in\mathbb{R}^d$. The margins of $G_{\star,j}$ of $G_\star$ are unit Fréchet distributed *i.e.*

$$\forall x>0,\ G_{\star,j}(x)=\exp\{-1/x\}. \tag{A.17}$$

As a consequence, it is possible to characterize a GEVD with two components: On the one hand, the marginal tail indices $\boldsymbol{\gamma}$, which characterize its margins and, on the other hand, its dependence structure encoded in $G_\star$.

### A.1.3 Dependence structure

The following result establishes a strong link between the stable tail dependence function (stdf) of $G$ (as introduced in Definition 2.1) and its simple max-stable cdf $G_\star$.

**Proposition A.4** ([HF06, Section 6.1.5]). *Consider a max-stable cdf $G$ and its associated simple max-stable cdf $G_\star$ defined in (A.16). The stdf of $G$ is given by*

$$\forall \boldsymbol{x}\in(0,\infty)^d:\ \ell_G(\boldsymbol{x})=-\log G_\star(1/\boldsymbol{x}).$$

We show in the next paragraph that both $\ell_G$ and $G_\star$ can be interpreted in terms of D-norms, which is a key property for our analysis.

## A.2 D-norms

Let us first recall classical results from multivariate extreme-value theory through the lens of D-norms. The reference monograph on the matter is [Fal19].

### A.2.1 Definition and basic properties

We start with the basic definitions.

**Definition A.5** (Norm and D-norms). A function $\|\cdot\|:\mathbb{R}^d\to\mathbb{R}^+$ is a norm on $\mathbb{R}^d$ if it verifies three conditions: a) *Positive definiteness*, b) *Absolute Homogeneity*, c) *Triangle inequality.* A subset of norms is the set of D-norms defined by

$$\|\cdot\|_{\boldsymbol{\Gamma}}:\ \boldsymbol{x}\in\mathbb{R}^d\mapsto\|\boldsymbol{x}\|_{\boldsymbol{\Gamma}}=\mathbb{E}\left[\max_{i\in[d]}(|x_i|\boldsymbol{\Gamma}_i)\right]$$

for a non-negative random variable $\boldsymbol{\Gamma}\in\mathbb{R}^d$ with unit expectation (Definition 2.3), $\boldsymbol{\Gamma}$ is also known as the generator of the D-norm.

Let us give a few remarks. First, since D-norms are monotone and radially symmetric, not all norms are D-norms (see [Fal19, Chapter 1]). Second, usual norms (such as the $L_p$-norms for $p \in [1, \infty]$) are D-norms. Last, observe that there are infinitely many generators $\mathbf{\Gamma}$ leading to the same D-norm: multiply $\mathbf{\Gamma}$ by an independent positive random variable $\boldsymbol{U}$ with unit expectation, it readily gives $\|\cdot\|_{\mathbf{\Gamma}\boldsymbol{U}} = \|\cdot\|_{\mathbf{\Gamma}}$. Hence, to ease notations, sometimes we may prefer to write the above norm by $\|\cdot\|_D$ to focus less on the generator $\mathbf{\Gamma}$; in such a situation, we refer to the generator $\mathbf{\Gamma}$ associated with the D-norm by writing $\mathbf{\Gamma} \triangleleft \|\cdot\|_D$, which reads "$\mathbf{\Gamma}$ generates $\|\cdot\|_D$". Despite the multiple generators associated with the same D-norm, the following theorem provides a uniqueness result on the choice of $\mathbf{\Gamma}$ when the norm of the random variable $\mathbf{\Gamma}$ is constrained.

**Theorem A.6** (Normed Generators, [Fal19, Theorem 1.7.1])**.** *Let $\|\cdot\|$ be an arbitrary norm on $\mathbb{R}^d$. For any D-norm $\|\cdot\|_D$ on $\mathbb{R}^d$, there exists a generator $\mathbf{\Gamma} \triangleleft \|\cdot\|_D$ and a constant $c$ with the additional property $\mathbb{P}(\|\mathbf{\Gamma}\| = c) = 1$. The distribution of the generator is uniquely defined.*

Let us remark that taking $\|\cdot\| = \|\cdot\|_1$ as the $L_1$-norm, the constant $c$ is necessarily $d$:

$$\|\mathbf{\Gamma}\|_1 = c \ \ a.s. \quad \Rightarrow \ c = \mathbb{E}\left[\|\mathbf{\Gamma}\|_1\right] = \sum_{i \in [d]} \mathbb{E}\left[\mathbf{\Gamma}_i\right] = d.$$

### A.2.2 Relation with stdf

The next result is fundamental to connect a stdf to a D-norm.

**Theorem A.7** (Representation of simple max stable distributions)**.** *A cdf $G_\star$ on $\mathbb{R}^d$ is simple max-stable if and only if there exists a D-norm $\|\cdot\|_D$ on $\mathbb{R}^d$ such that:*

$$\forall \boldsymbol{x} \in (0, \infty)^d : \ G_\star(\boldsymbol{x}) = \exp\left\{ - \left\| \frac{\mathbf{1}}{\boldsymbol{x}} \right\|_D \right\}. \tag{A.18}$$

*In particular, if $G$ is a max-stable cdf on $\mathbb{R}^d$, then its stdf is given by*

$$\ell_G(\boldsymbol{x}) = \|\boldsymbol{x}\|_D. \tag{A.19}$$

The first statement (A.18) is taken from [Fal19, Theorem 2.3.3] or [FF21, Theorem 4.1], while the second one (A.19) follows from Proposition A.4.

## A.3 Quantization

**The quantization problem.** The quantization problem is concerned with finding the best approximation of a random vector $\boldsymbol{X} \in \mathbb{R}^d$ with a quantized version, *i.e.* a version that takes a discrete number of values. For a general reference on quantization, see [GL00]. Finding the best approximation of a random vector depends on the chosen measure. Define the quantization problem [GL00, Section 10.1] of a probability measure $P$, for a given norm $\|\cdot\|$ (for example the Euclidean norm), as:

$$\begin{aligned} e_{N,\infty}(P) &:= \inf_{f : \mathbb{R}^d \to \mathbb{R}^d} \left\{ P\text{-esssup} \, \|\boldsymbol{X} - f(\boldsymbol{X})\| : \#f(\mathbb{R}^d) \leq N \right\} \\ &= \inf_{\substack{\boldsymbol{\alpha} \subset \mathbb{R}^d \\ \#\boldsymbol{\alpha} \leq N}} \ \sup_{\boldsymbol{x} \in \mathrm{supp}(P)} \ \min_{\boldsymbol{a} \in \boldsymbol{\alpha}} \|\boldsymbol{x} - \boldsymbol{a}\|, \end{aligned}$$

where $f : \mathbb{R}^d \to \mathbb{R}^d$ is a quantizer. Since $e_{N,\infty}(P)$ depends on the probability measure $P$ only through its support, sometimes (see Lemma A.9 below) we will write $e_{N,\infty}(A)$ as a function of a set $A \subset \mathbb{R}^d$ (that can be taken as the support of $P$).
Then, if $\text{supp}(P)$ is compact, Jordan measurable with positive volume [GL00, p.3], the following limit exists:

$$Q_\infty\left(\text{supp}(P)\right) = \lim_{N\to\infty} N^{1/d} e_{N,\infty}(P). \tag{A.20}$$

Note that $Q_\infty\left(\text{supp}(P)\right)$ does not depend on the law of $P$ but only on its support. Therefore, we will interchangeably use a probability distribution or a set as input of $Q_\infty(\cdot)$. The limit $Q_\infty\left(\text{supp}(P)\right)$ is called the covering coefficient or quantization coefficient of order $\infty$ and can be expressed in terms of the covering coefficient $Q_\infty\left([0,1]^d\right)$. Moreover, with a further constraint, we have the following theorem:

**Theorem A.8** ([GL00, Theorem 10.7])**.** *Let $A \subset \mathbb{R}^d$ be a non empty compact set with $\lambda^d(\partial A) = 0$. Let $Q_\infty\left([0,1]^d\right)$ be the quantization coefficient of $[0,1]^d$ defined in (A.20). Then $Q_\infty\left([0,1]^d\right) > 0$ and:*

$$\lim_{N\to\infty} N^{1/d} e_{N,\infty}(A) = Q_\infty\left([0,1]^d\right) \lambda^d(A)^{1/d}, \tag{A.21}$$

where $\lambda^d$ is the Lebesgue measure in dimension $d$. We state the following technical lemma, which will be helpful for the proof of Theorem 2.12:

**Lemma A.9** ([GL00, Lemma 10.6])**.**

1. *Let $A, B \subset \mathbb{R}^d$ be nonempty compact sets with $A \subset B$. Then,*

$$\forall N \in \mathbb{N} : \; e_{N,\infty}(A) \leq e_{N,\infty}(B). \tag{A.22}$$

2. *Let $N$, $m$ and $\{N_k\}_{k\in[m]}$ be positive integers such that $\sum_{k\in[m]} N_k \leq N$ and consider $m$ nonempty compact sets $\{A_k\}_{k\in[m]}$. Then,*

$$e_{N,\infty}\left(\bigcup_{k\in[m]} A_k\right) \leq \max_{k\in[m]} e_{N_k,\infty}\left(A_k\right). \tag{A.23}$$

Here is a particular case of (A.23) when $N \geq 2m$:

$$e_{N,\infty}\left(\bigcup_{k\in[m]} A_k\right) \leq \max_{k\in[m]} e_{\lceil N/m\rceil - 1,\infty}\left(A_k\right). \tag{A.24}$$

*Proof.* The proof of (A.23) can be found in [GL00, Lemma 10.6]. Equation (A.24) is deduced from Equation (A.23) with $N_k = \lceil N/m\rceil - 1$ for all $k \in [m]$. □

## A.4 Copulas, measures of dependence

**Copulas.** Let us consider a $d$-variate random vector $\boldsymbol{X}$ from a cdf $F$ with continuous margins $F_j$. Sklar's Theorem [Skl59] states that there exists a unique function $C : [0,1]^d \to [0,1]$ such that, for all $\boldsymbol{x} \in \mathbb{R}^d$,

$$F(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d)).$$

Introducing for any $j \in [d]$ the uniformly distributed random variable $U_j = F_j(x_j)$, the copula $C$ is the cdf of the joint random vector $(U_1, \dots, U_d)$. Copulas are a tool for studying dependence of probability objects, independently from the margins. See [Nel06] for a thorough overview. As an example, the Gumbel copula is given by:

$$\forall \boldsymbol{u} \in [0,1]^d : \; C_\beta(\boldsymbol{u}) = \exp\left\{ -\left( \sum_{i \in [d]} (-\log(u_i)^\beta \right)^{1/\beta} \right\},$$

where $\beta \geq 1$, see [Nel06, Equation (2.4.2.)].

**Definition A.10** (Kendall's $\tau$, [Nel06, Theorem 5.1.3.])**.** Let $C$ be a bivariate copula. Its associated Kendall's $\tau$ is:

$$\tau = 4\mathbb{E}\left[C(U_1, U_2)\right] - 1, \tag{A.25}$$

where $(U_1, U_2) \sim C$.

Kendall's $\tau$ is a common measure of dependence for copulas, see [Joe14, Section 6.8] for an estimation procedure. Note that Kendall's $\tau$ for the Gumbel copula is given by $\tau = 1 - 1/\beta$, $\beta \geq 1$.

**Wasserstein distance.** Let $\mu$ and $\nu$ be two measures on $\mathbb{R}^d$. Considering a cost function $c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ and $p \geq 1$, the $p$ - Wasserstein distance associated with the cost $c$ between $\mu$ and $\nu$ is defined as [Vil09, Chap. 6]:

$$\mathcal{W}_p(\mu, \nu) = \left( \inf_{\gamma \in \Gamma(\mu,\nu)} \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y}) \sim \gamma} \left[ c(\boldsymbol{X}, \boldsymbol{Y})^p \right] \right)^{1/p},$$

where $\Gamma(\mu, \nu)$ is the set of *couplings* of $\mu$ and $\nu$, *i.e.* the set of distributions on $\mathbb{R}^d \times \mathbb{R}^d$ such that the marginals are respectively $\mu$ and $\nu$. The Wasserstein distance suffers from the fact that there exists no closed formula to compute it, nor very accessible computable estimates in dimensions higher that one. In order to perform estimation, we therefore rely on the sliced-Wasserstein distance, which is an efficient proxy, much used for computations in experimental settings [Nad21]. It is computed by randomly projecting the two compared measures on axes defined by directions $\boldsymbol{v}$ and averaging over the projections. Specifically, the sliced-Wasserstein distance is defined as:

$$\mathrm{SWD}_p(\mu, \nu) = \mathop{\mathbb{E}}_{\boldsymbol{v} \sim \mathcal{U}\left(\mathbb{S}^{d-1}\right)} \left( \mathcal{W}_p^p \left( \boldsymbol{v}_{\#}\mu, \boldsymbol{v}_{\#}\nu \right) \right)^{1/p}, \tag{A.26}$$

where $\mathbb{S}^{d-1}$ is the Euclidean unit sphere, $\boldsymbol{v}_{\#}\mu$ (respectively $\boldsymbol{v}_{\#}\nu$) stands for the pushforward of the projection $\boldsymbol{X} \in \mathbb{R}^d \mapsto \langle \boldsymbol{X}, \boldsymbol{v} \rangle \in \mathbb{R}$, where $\boldsymbol{X} \sim \mu$ (respectively $\nu$).

# B Proofs of claims

## B.1 Proof of Proposition 2.10

The following states an interesting corollary of Theorem 2.8, which we are the first to prove formally as far as we know.

**Corollary B.1** (of Theorem 2.8)**.** *Let $\boldsymbol{Z} \in \mathbb{R}^N$ be a positive random vector with continuous cdf $F_{\boldsymbol{Z}} \in \mathrm{Dom}\left(G_{\boldsymbol{\Gamma}_{\boldsymbol{Z}}}\right)$ where $G_{\boldsymbol{\Gamma}_{\boldsymbol{Z}}}$ is a simple max-stable distribution, i.e. $F_{\boldsymbol{Z}}$ satisfies (2.4) and (2.5). Consider $\Phi \in \mathbb{H}_1\left([0,\infty)^N, [0,\infty)^d\right)$, such that:*

$$\mathbb{E}\left[\Phi\left(\boldsymbol{\Gamma}_{\boldsymbol{Z}}\right)\right] > \mathbf{0}.$$

*Let $\boldsymbol{X} = \Phi(\boldsymbol{Z})$. Then, $\Phi\left(\boldsymbol{\Gamma}_{\boldsymbol{Z}}\right) / \mathbb{E}\left[\Phi\left(\boldsymbol{\Gamma}_{\boldsymbol{Z}}\right)\right] =: \boldsymbol{\Gamma}_{\boldsymbol{X}}$ is a valid generator of a D-norm and $F_{\boldsymbol{X}} \in \mathrm{Dom}\left(G_{\boldsymbol{\Gamma}_{\boldsymbol{X}}}\right)$ where $G_{\boldsymbol{\Gamma}_{\boldsymbol{X}}}$ is a simple max-stable distribution. Moreover, the normalizing constants to ensure convergence in (A.14) are $\boldsymbol{a}_t = t\,\mathbb{E}\left[\Phi\left(\boldsymbol{\Gamma}_{\boldsymbol{Z}}\right)\right]$ and $\boldsymbol{b}_t = \mathbf{0}$.*

*Proof.* In view of the characterization of Theorem 2.8, consider any $h \in \mathbb{H}_1\left([0,\infty)^d, [0,\infty)\right)$ and

$$\Psi_h : \left\{ \begin{array}{ccc} [0,\infty)^N & \to & [0,\infty), \\ \boldsymbol{z} & \mapsto & h\left(\frac{\Phi(\boldsymbol{z})}{\mathbb{E}[\Phi(\boldsymbol{\Gamma}_{\boldsymbol{Z}})]}\right). \end{array} \right.$$

Clearly, $\Psi_h$ is continuous 1-homogeneous as composition of the continuous 1-homogeneous functions $\Phi$ and $h$. Since $F_{\boldsymbol{Z}}$ verifies (2.5), it follows that:

$$t\,\mathbb{P}\left(h\left(\frac{\Phi(\boldsymbol{Z})}{\mathbb{E}\left[\Phi(\boldsymbol{\Gamma}_{\boldsymbol{Z}})\right]}\right) > t\right) = t\,\mathbb{P}\left(\Psi_h(\boldsymbol{Z}) > t\right) \underset{t\to\infty}{\longrightarrow} \mathbb{E}\left[\Psi_h(\boldsymbol{\Gamma}_{\boldsymbol{Z}})\right] = \mathbb{E}\left[h\left(\frac{\Phi(\boldsymbol{\Gamma}_{\boldsymbol{Z}})}{\mathbb{E}\left[\Phi(\boldsymbol{\Gamma}_{\boldsymbol{Z}})\right]}\right)\right].$$

Introducing $\boldsymbol{Y} = \Phi(\boldsymbol{Z})/\mathbb{E}\left[\Phi(\boldsymbol{\Gamma}_{\boldsymbol{Z}})\right] = \boldsymbol{X}/\mathbb{E}\left[\Phi(\boldsymbol{\Gamma}_{\boldsymbol{Z}})\right]$ and its corresponding distribution function $F_{\boldsymbol{Y}}$, the previous convergence can be rewritten as:

$$\forall h \in \mathbb{H}_1\left([0,\infty)^d, [0,\infty)\right) : \quad t\,\mathbb{P}\left(h\left(\boldsymbol{Y}\right) > t\right) \underset{t\to\infty}{\longrightarrow} \mathbb{E}\left[h\left(\boldsymbol{\Gamma}_{\boldsymbol{X}}\right)\right],$$

so that $\boldsymbol{Y}$ satisfies characterization (2.5) in Theorem 2.8. Moreover, since $\mathbb{E}\left[\boldsymbol{\Gamma}_{\boldsymbol{X}}\right] = \mathbf{1}$ and the coordinates of $\boldsymbol{\Gamma}_{\boldsymbol{X}}$ are non-negative, $\boldsymbol{\Gamma}_{\boldsymbol{X}}$ is indeed the generator of a D-norm. Therefore:

$$\forall \boldsymbol{x} \in (0,\infty)^d : \quad F_{\boldsymbol{Y}}^t(t\boldsymbol{x}) \underset{t\to\infty}{\longrightarrow} \exp\left\{-\left\|\frac{1}{\boldsymbol{x}}\right\|_{\boldsymbol{\Gamma}_{\boldsymbol{X}}}\right\}.$$

Noting that $F_{\boldsymbol{Y}}(\boldsymbol{x}) = F_{\boldsymbol{X}}(\mathbb{E}\left[\Phi(\boldsymbol{\Gamma}_{\boldsymbol{Z}})\right]\boldsymbol{x})$ finally proves

$$\forall \boldsymbol{x} \in (0,\infty)^d : \quad F_{\boldsymbol{X}}^t(t\,\mathbb{E}\left[\Phi(\boldsymbol{\Gamma}_{\boldsymbol{Z}})\right]\boldsymbol{x}) \underset{t\to\infty}{\longrightarrow} \exp\left\{-\left\|\frac{1}{\boldsymbol{x}}\right\|_{\boldsymbol{\Gamma}_{\boldsymbol{X}}}\right\} =: G_{\boldsymbol{\Gamma}_{\boldsymbol{X}}}(\boldsymbol{x}),$$

and the result is proven. □

We now complete the proof of Proposition 2.10. The previous result holds for continuous 1-homogeneous function $\Phi$, which is not exactly the situation with a neural network with ReLU activation functions because of the bias terms (see Proposition 2.9). However, we claim that, asymptotically, these bias terms do not play any role on the stdf. To see this, let us investigate the behavior of a ReLU-neural network when going from a layer to the next one. Denoting by $\sigma$ the ReLU activation function, and supposing that $\varphi_l = \sigma$, $\forall l \in [L]$, we have

$$\begin{aligned} \boldsymbol{Z}_l &= \sigma\left(\boldsymbol{W}_l \boldsymbol{Z}_{l-1} + \boldsymbol{b}_l\right) \\ &= \sigma\left(\boldsymbol{W}_l \boldsymbol{Z}_{l-1}\right) + \left[\sigma\left(\boldsymbol{W}_l \boldsymbol{Z}_{l-1} + \boldsymbol{b}_l\right) - \sigma\left(\boldsymbol{W}_l \boldsymbol{Z}_{l-1}\right)\right]. \end{aligned} \tag{B.27}$$

On the one hand, $\sigma\left(\boldsymbol{W}_l \boldsymbol{Z}_{l-1}\right)$ is a 1-homogeneous function of $\boldsymbol{Z}_{l-1}$ and is thus a heavy-tailed random variable if $\boldsymbol{W}_l \neq \mathbf{0}$ and $\boldsymbol{Z}_{l-1}$ is heavy-tailed itself. On the other hand, the

remaining term $\left[\cdots\right]$ in (B.27) is a bounded random variable (bounded by $|\boldsymbol{b}_l|$); the stdf of the sum of a heavy-tailed random variable and a bounded one is the stdf of the heavy-tailed random variable. The stdf of (B.27) is therefore the stdf of the transformation without the bias, and by induction the stdf of the output is the stdf of the input after iterative applications of $\sigma\left(\boldsymbol{W}_l\,\cdot\right)$ for $l \in [L]$, which is 1-homogeneous continuous. Let denote by $\Phi$ the output of the $L$ compositions of $\sigma\left(\boldsymbol{W}_l\,\cdot\right)$: since the matrix weights $\boldsymbol{W}_l$ are non-negative, $\Phi \in \mathbb{H}_1\left([0,\infty)^N, [0,\infty)^d\right)$. Therefore, Corollary B.1 applies and so the stdf associated with the output of the network is $\boldsymbol{\Gamma}_{\mathcal{G}_{\boldsymbol{Z}}} = \Phi(\boldsymbol{\Gamma}_{\boldsymbol{Z}})/\mathbb{E}\Phi(\boldsymbol{\Gamma}_{\boldsymbol{Z}})$, which has at most $N$ atoms. Thus, the proof is concluded.

## B.2 Proof of Proposition 2.11

The next lemma can be interpreted in the following way. If a discrete random vector generates a D-norm, then it admits a discrete counterpart which can be any other discrete distribution with modified atoms: it is also a generator of the concerned D-norm.

**Lemma B.2.** *Let $\boldsymbol{\Gamma} \triangleleft \|\cdot\|_D$ where $\boldsymbol{\Gamma} \in [0,\infty)^d$ is discrete, that is $\boldsymbol{\Gamma} \sim \mu_{\boldsymbol{\Gamma}}$ where*

$$\mu_{\boldsymbol{\Gamma}} = \sum_{i\in[N]} p_i \delta_{\gamma^{(i)}}.$$

*Consider a new discrete probability measure with positive weights $(q_i)_{i\in[N]}$ and a new random variable $\widetilde{\boldsymbol{\Gamma}} \sim \mu_{\widetilde{\boldsymbol{\Gamma}}}$, where*

$$\mu_{\widetilde{\boldsymbol{\Gamma}}} := \sum_{i\in[N]} q_i \delta_{\widetilde{\gamma}^{(i)}},$$

*with $\widetilde{\gamma}^{(i)} = \frac{p_i}{q_i}\gamma^{(i)}$, $i \in [N]$. Then, $\widetilde{\boldsymbol{\Gamma}}$ is a valid generator and $\widetilde{\boldsymbol{\Gamma}} \triangleleft \|\cdot\|_D$.*

*Proof.* By definition of the D-norm, one has $\forall \boldsymbol{x} \in \mathbb{R}^N$:

$$\begin{aligned}
\|\boldsymbol{x}\|_D &= \mathbb{E}_{\boldsymbol{\Gamma}}\left[\max_{j\in[N]}\left\{|x_j|\boldsymbol{\Gamma}_j\right\}\right] \\
&= \sum_{i\in[N]} p_i \max_{j\in[N]}\left\{|x_j|\gamma_j^{(i)}\right\} \\
&= \sum_{i\in[N]} q_i \max_{j\in[N]}\left\{|x_j|\widetilde{\gamma}_j^{(i)}\right\} \\
&= \mathbb{E}_{\widetilde{\boldsymbol{\Gamma}}}\left[\max_{j\in[N]}\left\{|x_j|\widetilde{\boldsymbol{\Gamma}}_j\right\}\right].
\end{aligned}$$

Since $\mathbb{E}_{\widetilde{\boldsymbol{\Gamma}}}\left[\widetilde{\boldsymbol{\Gamma}}\right] = \mathbb{E}_{\boldsymbol{\Gamma}}\left[\boldsymbol{\Gamma}\right] = \mathbf{1}$ and $\widetilde{\boldsymbol{\Gamma}} \in [0,\infty)^d$, $\widetilde{\boldsymbol{\Gamma}}$ is a valid generator. □

In particular, it is possible to choose uniform weights, $q_i = 1/N$, $i \in [N]$. In other words, if a D-norm admits a generator which admits a finite number of values, it also admits a generator with the same number of finite values, with uniform probability. This result is important as it means that the class of generators of D-norms which take a finite number of values and with uniform weights is as rich as the class of generator which take finite values with arbitrary weights.

**Proposition B.3.** *Consider a generator $\mathbf{\Gamma}$ of a D-norm in $\mathbb{R}^d$. Suppose that it is discrete uniform with $N$ atoms $(\gamma^{(i)})_{i\in[N]}$:*

$$\mu_{\mathbf{\Gamma}} = \frac{1}{N}\sum_{i\in[N]} \delta_{\gamma^{(i)}}. \tag{B.28}$$

*Consider morever a random vector $\boldsymbol{Z} \in \mathbb{R}^N$ with i.i.d. unit Fréchet margins: $\boldsymbol{Z} = (Z_1, \dots, Z_N)$. Then, there exists a linear mapping $\Phi : \mathbb{R}^N \to \mathbb{R}^d$ such that:*

$$F_{\Phi(\boldsymbol{Z})} \in \mathrm{Dom}\left(G_{\mathbf{\Gamma}}\right).$$

*Proof.* Let us first remark that $F_{\boldsymbol{Z}} \in \mathrm{Dom}\left(G_{\mathbf{\Gamma}_{\boldsymbol{Z}}}\right)$ with:

$$\mu_{\mathbf{\Gamma}_{\boldsymbol{Z}}} = \frac{1}{N}\sum_{i\in[N]} \delta_{N\boldsymbol{e}_i},$$

see Example 2.6. Consider $\mathbf{\Gamma} \in \mathbb{R}^N$ a valid generator of a D-norm with a discrete uniform distribution (B.28). There exists $\Phi : \mathbb{R}^N \to \mathbb{R}^d$ a linear function such that $\forall i \in [N] :\ \Phi(N\boldsymbol{e}_i) = \gamma^{(i)}$. Let us note that $\Phi$ is continuous $1-$homogeneous because it is linear. Moreover, remark that $\forall i \in [N] :\ \gamma^{(i)} \in [0,\infty)^d$ implies $\Phi\left([0,\infty)^N\right) \subset \Phi\left([0,\infty)^d\right)$ and thus $\Phi(\boldsymbol{Z}) \in [0,\infty)^d$. Corollary B.1 ensures that

$$F_{\Phi(\boldsymbol{Z})} \in \mathrm{Dom}\left(G_{\Phi(\mathbf{\Gamma}_{\boldsymbol{Z}})/\mathbb{E}(\Phi(\mathbf{\Gamma}_{\boldsymbol{Z}}))}\right) = \mathrm{Dom}\left(G_{\mathbf{\Gamma}}\right)$$

since, as $\Phi\left(N\boldsymbol{e}_i\right) = \gamma^{(i)}$, $\Phi(\mathbf{\Gamma}_{\boldsymbol{Z}})$ is equal in distribution to $\mathbf{\Gamma}$ (note that $\mathbb{E}\left[\Phi(\mathbf{\Gamma}_{\boldsymbol{Z}})\right] = \mathbf{1}$). The result is thus proved. □

**Remark B.4.** *This result ensures that, given any random vector of interest $\boldsymbol{X}$ for which the associated D-norm generator $\mathbf{\Gamma}_{\boldsymbol{X}}$ has a uniform distribution with $N$ atoms, one can find a linear map $\Phi$ transforming a noise with unit Fréchet margins with at least $N$ atoms to a distribution whose D-norm generator is the desired one.*

Lemma B.2 and Proposition B.3 together prove Proposition 2.11.

## B.3 Proof of Theorem 2.12

The following technical lemma will be useful to prove the result:

**Lemma B.5.** *$\forall \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^d$:*

$$\left|\max_{i\in[d]} a_i b_i - \max_{i\in[d]} a_i c_i\right| \leq \|\boldsymbol{a}\|_\infty \max_{i\in[d]} |b_i - c_i|.$$

*Proof.* Without loss of generality, suppose that $\max_{i\in[d]} a_i b_i \geq \max_{i\in[d]} a_i c_i$, the argument being symmetric. One has:

$$\begin{aligned}
\left|\max_{i\in[d]} a_i b_i - \max_{i\in[d]} a_i c_i\right| &= \max_{i\in[d]}\left\{a_i c_i + a_i b_i - a_i c_i\right\} - \max_{i\in[d]} a_i c_i \\
&\leq \max_{i\in[d]} a_i c_i + \max_{i\in[d]}\left\{a_i b_i - a_i c_i\right\} - \max_{i\in[d]} a_i c_i \\
&\leq \max_{i\in[d]} |a_i| \max_{i\in[d]} |b_i - c_i|
\end{aligned}$$

and the result is proved.

□

We seek to find a upper bound on:

$$\inf_{\theta\in\Theta}\sup_{x\in(0,\infty)^d}\left|\ell_F(\boldsymbol{x})-\ell_{G_{\theta,N}}(\boldsymbol{x})\right|/\left\|\boldsymbol{x}\right\|_\infty.$$

In view of the above, and remembering that the stdf of the output of a neural network is discrete (Proposition 2.10), our goal is to find the best approximation of the D-norm $\|.\|_{\boldsymbol{\Gamma}}$ with a discretized generator with $N$ atoms; we will leverage results from quantization (Section A.3). Take $\boldsymbol{\Gamma}$ an arbitrary generator of the D-norm related to $\ell_F$. One has:

$$\begin{aligned}\min_{f:\mathbb{R}^d\to\mathbb{R}^d,\#f(\mathbb{R}^d)=N}\left|\left\|\boldsymbol{x}\right\|_{\boldsymbol{\Gamma}}-\left\|\boldsymbol{x}\right\|_{f(\boldsymbol{\Gamma})}\right| &= \min_{f:\mathbb{R}^d\to\mathbb{R}^d,\#f(\mathbb{R}^d)=N}\left|\mathbb{E}_{\boldsymbol{\Gamma}}\left[\max_{i\in[d]}(|x_i|\boldsymbol{\Gamma}_i)-\max_{i\in[d]}(|x_i|f(\boldsymbol{\Gamma}_i))\right]\right|,\\ &\leq \min_{f:\mathbb{R}^d\to\mathbb{R}^d,\#f(\mathbb{R}^d)=N}\left\|\boldsymbol{x}\right\|_\infty\mathbb{E}_{\boldsymbol{\Gamma}}\left[\max_{i\in[d]}|\boldsymbol{\Gamma}_i-f(\boldsymbol{\Gamma}_i)|\right] \quad (B.29)\\ &\leq \left\|\boldsymbol{x}\right\|_\infty e_{N,\infty}(\boldsymbol{\Gamma}), \quad (B.30)\end{aligned}$$

from Lemma B.5. To bound (B.30), we use a quantizer which has a constant sup-norm, which existence is guaranteed by (Theorem A.6):

$$\left\|\boldsymbol{\Gamma}\right\|_\infty = c \quad a.s.. \quad (B.31)$$

Observe that the constant $c$ must be related to the stdf using Definition 2.3:

$$\ell_F(1)=\mathbb{E}\left[\left\|\boldsymbol{\Gamma}\right\|_\infty\right]=c.$$

Now, we are in a position to derive a bound on $e_{N,\infty}(\boldsymbol{\Gamma})$.

**Definition B.6.** For any norm $\|\cdot\|$, let us denote by $\mathcal{B}_{\|\cdot\|}$ the associated unit ball. Besides, let $\mathcal{B}^+_{\|\cdot\|}=\mathcal{B}_{\|\cdot\|}\cap[0,\infty)^d$ be the part of the unit ball in the positive orthant.

Owing to (B.31) and Definition 2.3, $\operatorname{supp}(\boldsymbol{\Gamma})\subset c\cdot[0,1]^d$. Equation (A.21) shows that one can get an error bound for (B.30) with rate $N^{-1/d}$. However, it is possible to improve this result by remarking that $\boldsymbol{\Gamma}$ takes values on the boundary of a hypercube (see (B.31)). Indeed, now note that:

$$\operatorname{supp}(\boldsymbol{\Gamma})\subset c\cdot\mathcal{B}^+_{\|\cdot\|_\infty}\subset\bigcup_{i\in[d]}c\cdot\mathcal{B}^+_{\|\cdot\|_\infty}(i), \quad (B.32)$$

where, for all $i\in[d]$,

$$\mathcal{B}^+_{\|\cdot\|_\infty}(i)=\left\{(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_d):(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_d)\in[0,1]^{d-1}\right\}.$$

**Remark B.7.** *For all $i\in[d]$, $\mathcal{B}^+_{\|\cdot\|_\infty}(i)$ is canonically homeomorph to $[0,1]^{d-1}$ in the topological space $\mathbb{R}^{d-1}$ endowed with the topology of $\|\cdot\|_\infty$.*

Using Lemma A.9 (inequality (A.22) with (B.32) from first to second term and from second to third and inequality (A.23) from third to fourth), we get

$$e_{N,\infty}\left(\boldsymbol{\Gamma}\right)\leq c\cdot e_{N,\infty}\left(\mathcal{B}^+_{\|\cdot\|_\infty}\right)\leq c\cdot e_{\lceil N/d\rceil-1,\infty}\left(\mathcal{B}^+_{\|\cdot\|_\infty}(1)\right)=c\cdot e_{\lceil N/d\rceil-1,\infty}\left([0,1]^{d-1}\right).$$

Gathering previous arguments, we have proved the following:

**Lemma B.8.** *Consider $\mathbf{\Gamma} \in \mathbb{R}^d$ a generator of a D-norm such that $\|\mathbf{\Gamma}\|_\infty = c$ a.s. . Then, the following holds true:*

$$\forall \boldsymbol{x} \in \mathbb{R}^d : \min_{f:\mathbb{R}^d \to \mathbb{R}^d, \#f(\mathbb{R}^d)=N} \left| \|\boldsymbol{x}\|_{\mathbf{\Gamma}} - \|\boldsymbol{x}\|_{f(\mathbf{\Gamma})} \right| \leq c \cdot \|\boldsymbol{x}\|_\infty \, e_{\lceil N/d \rceil - 1, \infty} \left( [0,1]^{d-1} \right).$$

From (A.21), we get that:

$$\begin{aligned} e_{\lceil N/d \rceil - 1, \infty} \left( [0,1]^{d-1} \right) &\overset{N \to \infty}{\sim} (\lceil N/d \rceil - 1)^{-1/(d-1)} \, Q_\infty \left( [0,1]^{d-1} \right) \lambda([0,1]^{d-1})^{1/(d-1)} \\ &\overset{N \to \infty}{\sim} d^{1/(d-1)} N^{-1/(d-1)} Q_\infty \left( [0,1]^{d-1} \right) \end{aligned}$$

which proves (2.9), where $C(d) = c \cdot d^{1/(d-1)} Q_\infty \left( [0,1]^{d-1} \right)$ in (2.10). The proof of Theorem 2.12 is thus complete. ■

Observe that the quantization upper bound (B.30) is a bit rough since the right-hand side of (B.29) refers to the $L_1$-quantization that we bound using $L_\infty$-quantization. One could use $L_1$-quantization estimates but they would depend on the distribution of $\mathbf{\Gamma}$ and we prefer to provide worst-case estimates. Instead of using the $L_\infty$ and using $L_1$ quantization on (B.29), [GL00, Th. 6.2.] gives that the first order error of the error is proportional to the $L_{(d-1)/d}$ norm of the density of the generator on the boundaries of $\mathcal{B}^+_{\|\cdot\|_\infty}(i)$ times $N^{-1/(d-1)}$. In particular, if the distribution of the generator $\mathbf{\Gamma}$ (on the hypercube boundaries) has a null density with respect to the Lebesgue measure (*e.g.* if the distribution is discrete or singular), the multiplicative factor of the $N^{-1/(d-1)}$ term is null and the convergence of the bound can be quicker than $N^{-1/(d-1)}$.

## References


[AA10] S. Asmussen and H. Albrecher. *Ruin probabilities*. Advanced Series on Statistical Science & Applied Probability, 14. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, second edition, 2010.

[AGG22] M. Allouche, S. Girard, and E. Gobet. EV-GAN: Simulation of extreme events with ReLU neural networks. *Journal of Machine Learning Research*, 23(150):1–39, 2022.

[AGG24] M. Allouche, S. Girard, and E. Gobet. On the simulation of extreme events with neural networks. In Miguel de Carvalho, editor, *Handbook of Statistics of Extremes*, chapter 22. CRC Press, 2024. (to appear).

[BGTS04] J. Beirlant, Y. Goegebeur, J. Teugels, and J. Segers. *Statistics of extremes: theory and applications*. Wiley series in probability and statistics. Wiley, Hoboken, NJ, 2004.

[Buc04] J. A. Bucklew. *Introduction to rare event simulation*. Springer Series in Statistics. Springer-Verlag, New York, 2004.

[BZV+22] Y. Boulaguiem, J. Zscheischler, E. Vignotto, K. van der Wiel, and S. Engelke. Modeling and simulating spatial extremes by combining extreme value theory with generative adversarial networks. *Environmental Data Science*, 1:e5, 2022.

[CCXZ22] R. Cont, M. Cucuringu, R. Xu, and C. Zhang. Tail-GAN: Learning to simulate tail risk scenarios. *SSRN*, 2022.

[CSF+22] F. Cremer, B. Sheehan, M. Fortmann, A.N. Kia, M. Mullins, F. Murphy, and S. Materne. Cyber risk and cybersecurity: a systematic review of data availability. *The Geneva Papers on risk and insurance-Issues and practice*, 47(3):698–736, 2022.

[DG05] P. Del Moral and J. Garnier. Genealogical particle analysis of rare events. *The Annals of Applied Probability*, 15(4):2496–2534, 2005.

[ECB23] ECB Banking Supervision. 2023 stress test of euro area banks. *Available at https://www.bankingsupervision.europa.eu/ecb/pub/pdf/ssm.Report_2023_Stress_Test~96bb5a3af8.en.pdf*, ECB, 2023.

[EKM97] P. Embrechts, C. Klüppelberg, and T. Mikosch. *Modelling Extremal Events for Insurance and Finance*. Springer-Verlag, Berlin, 1997.

[Eur14] European Banking Authority. Guidelines on the revised common procedures and methodologies for the supervisory review and evaluation process (SREP) and supervisory stress testing. *Available at https://eba.europa.eu/regulation-and-policy/supervisory-review-and-evaluation-srep-and-pillar-2/guidelines-for-common-procedures-and-methodologies-for-the-supervisory-review-and-evaluation-process-srep-and-supervisory-stress-testing*, EBA/GL/2014/13, 2014.

[Fal19] M. Falk. *Multivariate Extreme Value Theory and D-Norms*. Springer Series in Operations Research and Financial Engineering. Springer International Publishing, Cham, 2019.

[FBS20] R. M. Feder, P. Berger, and G. Stein. Nonlinear 3D cosmic web simulation with heavy-tailed generative adversarial networks. *Physical Review D.*, 102(10):103504, 18, 2020.

[FF21] M. Falk and T. Fuller. New characterizations of multivariate Max-domain of attraction and D-Norms. *Extremes*, 24(4):849–879, 2021.

[FT28] R. A. Fisher and L. H. C. Tippett. Limiting forms of the frequency distribution of the largest or smallest member of a sample. *Mathematical Proceedings of the Cambridge Philosophical Society*, 24(2):180—190, 1928.

[GL00] S. Graf and H. Luschgy. *Foundations of quantization for probability distributions*. Number 1730 in Lecture Notes in Mathematics. Springer, Berlin; New York, 2000.

[GL15] E. Gobet and G. Liu. Rare event simulation using reversible shaking transformations. *SIAM Journal on Scientific Computing*, 37(5):A2295–A2316, 2015.

[Gne43] B. Gnedenko. Sur la distribution limite du terme maximum d'une série aléatoire. *Annals of Mathematics*, 44(3):423–453, 1943.

[GPAM+14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680, 2014.

[HCL+21] T. Huster, J. Cohen, Z. Lin, K. Chan, C. Kamhoua, N.O. Nandi, C-Y.J. Chiang, and V. Sekar. Pareto GAN: Extending the representational power of GANs to heavy-tailed distributions. In *International Conference on Machine Learning*, pages 4523–4532. PMLR, 2021.

[HEN+22] A. Hasan, K. Elkhalil, Y. Ng, J.M. Pereira, S. Farsiu, J. Blanchet, and V. Tarokh. Modeling extremes with $d$-max-decreasing neural networks. In *Uncertainty in Artificial Intelligence*, pages 759–768. PMLR, 2022.

[HF06] L. de Haan and Ana Ferreira. *Extreme value theory: an introduction*. Springer Series in Operations Research. Springer, New York ; London, 2006.

[HHP18] M. Hofert, R. Huser, and A. Prasad. Hierarchical archimax copulas. *Journal of Multivariate Analysis*, 167:195–211, 2018.

[HP24] T. Hickling and D. Prangle. Flexible Tails for Normalizing Flows, 2024. arXiv:2406.16971.

[Joe14] H. Joe. *Dependence modeling with copulas*. CRC press, 2014.

[KB14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014. arXiv:1412.6980.

[KW14] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *2nd International Conference on Learning Representation, ICLR*, 2014.

[LMH22] F. Liang, M. Mahoney, and L. Hodgkinson. Fat–Tailed Variational Inference with Anisotropic Tail Adaptive Flows. In *Proceedings of the 39th International Conference on Machine Learning*, pages 13257–13270. PMLR, 2022.

[MFE15] A.J. McNeil, R. Frey, and P. Embrechts. *Quantitative risk management*. Princeton Series in Finance. Princeton University Press, Princeton, NJ, 2015.

[MHN13] A.L. Maas, A.Y. Hannun, and A.Y. Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models. *Proceedings of the International Conference on Machine Learning (ICML)*, 2013.

[Mur22] K. P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[Nad21] K. Nadjahi. *Sliced-Wasserstein distance for large-scale machine learning : theory, methodology and extensions*. PhD thesis, Institut Polytechnique de Paris, November 2021.

[Nel06] R.B. Nelsen. *An introduction to copulas*. Springer series in statistics. Springer, New York Berlin Heidelberg, second edition, 2006.

[PNR+21] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

[PW05] M. Prandini and O. J. Watkins. Probabilistic aircraft conflict detection. *HYBRIDGE WP3: Reachability analysis for probabilistic hybrid systems*, 2005.

[Res87] S. I. Resnick. *Extreme Values, Regular Variation and Point Processes*. Springer Series in Operations Research and Financial Engineering. Springer, New York, NY, 1987.

[Rob03] P. Robert. *Stochastic networks and queues*, volume 52 of *Applications of Mathematics*. Springer-Verlag, Berlin, 2003.

[SDWMG15] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[Skl59] M. Sklar. Fonctions de répartition à $N$ dimensions et leurs marges. *Publications de l'Institut Statistique de l'Université de Paris*, (8):229–231, 1959.

[Vil09] C. Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

[WCZ+19] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng. Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. 17(1):26–40, 2019.

[WKKK20] M. Wiese, R. Knobloch, R. Korn, and P. Kretschmer. Quant GANs: deep generation of financial time series. *Quantitative Finance*, 20(9):1419–1440, 2020.

[ZMW+20] J. Zscheischler, O. Martius, S. Westra, E. Bevacqua, C. Raymond, R. M. Horton, B. van den Hurk, A. AghaKouchak, A. Jézéquel, M. D. Mahecha, D. Maraun, A. M. Ramos, N. N. Ridder, W. Thiery, and E. Vignotto. A typology of compound weather and climate events. *Nature Reviews Earth & Environment*, 1(7):333–347, 2020.