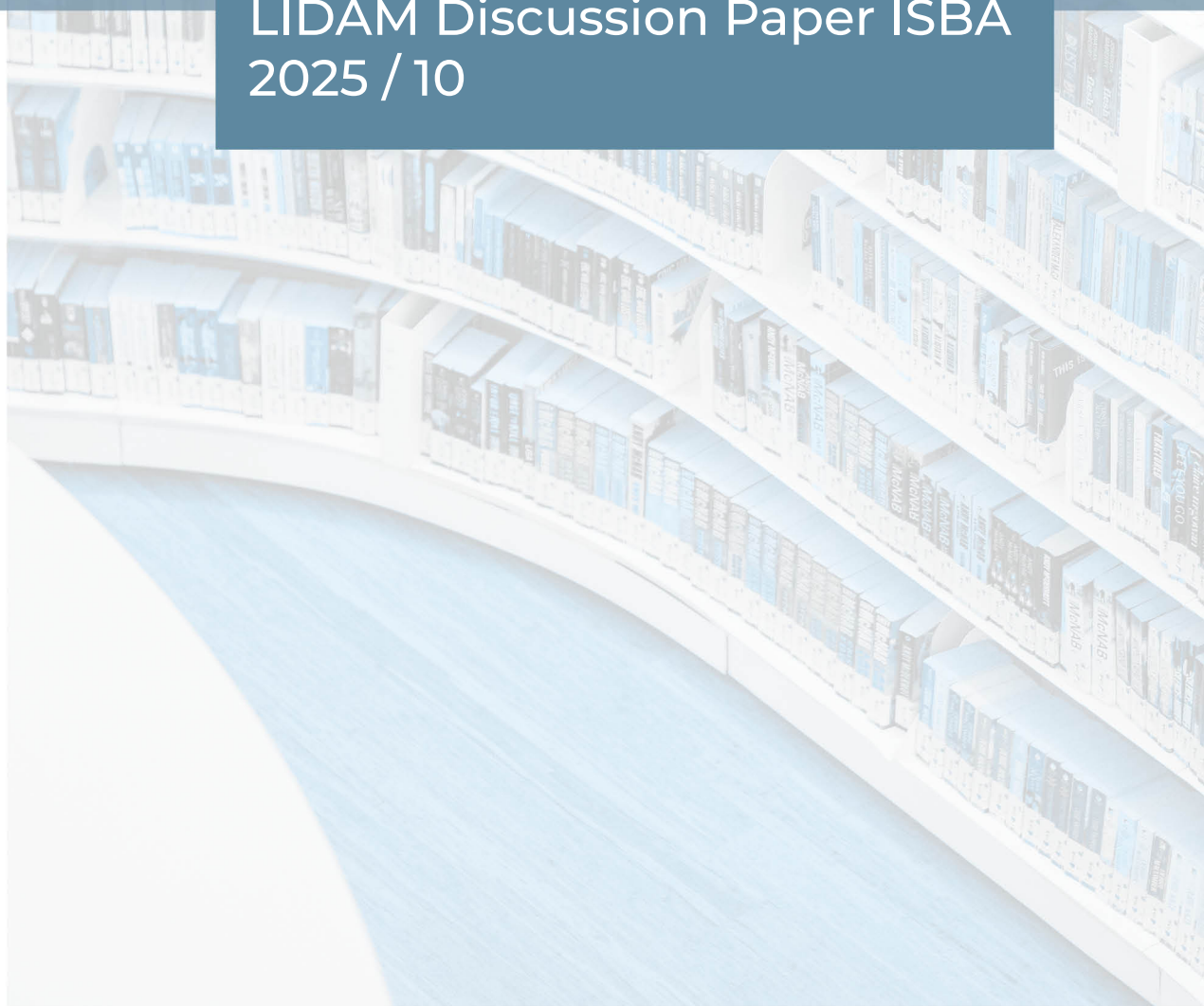
WASSERSTEIN–AITCHISON GAN FOR ANGULAR MEASURES OF MULTIVARIATE EXTREMES

Stéphane Lhaut, Holger Rootzén, Johan Segers

LIDAM Discussion Paper ISBA
2025 / 10



UCLouvain

ISBA | LIDAM
Louvain Institute of Data Analysis and Modeling in economics and statistics

ISBA

Voie du Roman Pays 20 - L1.04.01

B-1348 Louvain-la-Neuve

Email : lidam-library@uclouvain.be

https://uclouvain.be/en/research-institutes/lidam/isba/publication

Wasserstein–Aitchison GAN for angular measures of multivariate extremes

Stéphane Lhaut *[1], Holger Rootzén †[2], and Johan Segers ‡[1,3]

[1]Institute of Statistics, Biostatistics and Actuarial Sciences, LIDAM, UCLouvain, Voie du Roman Pays, 20, 1348 Louvain-La-Neuve, Belgium
[2]Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden
[3]Department of Mathematics, KU Leuven, Celestijnlaan 200B 02.28, 3001 Heverlee, Belgium

May 1, 2025

Abstract

Economically responsible mitigation of multivariate extreme risks – extreme rainfall in a large area, huge variations of many stock prices, widespread breakdowns in transportation systems – requires estimates of the probabilities that such risks will materialize in the future. This paper develops a new method, Wasserstein–Aitchison Generative Adversarial Networks (WA-GAN), which provides simulated values of future d-dimensional multivariate extreme events and which hence can be used to give estimates of such probabilities. The main hypothesis is that, after transforming the observations to the unit-Pareto scale, their distribution is regularly varying in the sense that the distributions of their radial and angular components (with respect to the L_1-norm) converge and become asymptotically independent as the radius gets large. The method is a combination of standard extreme value analysis modeling of the tails of the marginal distributions with nonparametric GAN modeling of the angular distribution. For the latter, the angular values are transformed to Aitchison coordinates in a full $(d-1)$-dimensional linear space, and a Wasserstein GAN is trained on these coordinates and used to generate new values. A reverse transformation is then applied to these values and gives simulated values on the original data scale. The method shows good performance compared to other existing methods in the literature, both in terms of capturing the dependence structure of the extremes in the data, as well as in generating accurate new extremes of the data distribution. The comparison is performed on simulated multivariate extremes from a logistic model in dimensions up to 50 and on a 30-dimensional financial data set.

Keywords: Extreme value theory, Wasserstein distance, Generative adversarial networks, Multivariate analysis, Aitchison coordinates

*stephane.lhaut@uclouvain.be, corresponding author
†hrootzen@chalmers.se
‡jjjsegers@kuleuven.be

1 Introduction

Dependence between extreme events is often of crucial importance. Extreme rainfall will be even more dangerous if it occurs at the same time at many spatial locations; costs associated with breakdowns in transportation systems are more harmful if they happen in many parts of the system; and extreme fluctuations in financial markets can cause much more damage if they occur in many parts of the economy. During the last decades the statistical research on dependent, or multivariate, extremes has grown rapidly and now contains many useful methods to model dependence between extreme values.

These methods typically build on parametric statistical models. However in the last three years papers using neural networks to simulate new multivariate extremes have started to appear. Often, but not always, these papers aim at nonparametric modeling of the dependence structure of the multivariate extreme values. The end result in the papers is an algorithm which can produce simulated values from the neural network "model" for the observations. A common approach is to transform the marginal distributions to some standard scale, then to combine nonparametric neural network modeling of the dependence structure with a parametric model from extreme value analysis (EVA) to simulate values on the standard scale, and finally to apply the inverse of the marginal transform to get simulated values on the real scale. Section 2.1 below gives a brief survey of some of this literature. We refer to [3] for a more extensive theoretical and simulation based overview of the area.

Here we develop a new method, WA-GAN, which combines parametric modeling of the marginal tails with nonparametric modeling of dependence in multivariate extremes. Our main assumptions are that the tails of the marginal distributions asymptotically follow a one-dimensional Generalized Pareto (GP) distribution and that after transformation to the unit-Pareto scale, the observations are multivariate regularly varying. Our method proceeds by the following main steps.

1. The d-dimensional observations are transformed to the unit-Pareto scale using the empirical marginal distribution functions.

2. Using the L_1-distance and a suitably large radius r, the transformed observations with radius larger than r are separated into a one-dimensional radial value and an angular value on the L_1-sphere.

3. Using Aitchison coordinates from compositional data, the angular values are transformed to values on the full $\mathbb{R}^{d-1}$ space.

4. A Wasserstein GAN is trained to use the Aitchison coordinates to produce simulated values on the L_1-sphere.

5. A result on multivariate GP distributions is used to transform the simulated values back to multivariate GP values on the original scale.

The next section contains the background needed for this together with a result (Proposition 1) providing the theoretical tool to transform the angular data of step 4

above to the real GP scale as described in step 5. Section 3 describes the details of WA-GAN, and provides the algorithms which are used by it. In Section 4.2 WA-GAN is compared to HTGAN [18] and GPGAN [28] on simulated data with a logistic dependence structure in dimension $d \in \{10, 50\}$ and, in Section 4.3, on financial data consisting of daily returns for $d = 30$ industry portfolios. Section 5 sums up the results of this paper.

2 Background

This section starts with a brief survey of the literature on neural network modeling of multivariate extremes. Section 2.2 gives background on regular variation and EVA and develops some new theory. Sections 2.3 and 2.4 contain brief recapitulations of Wasserstein GANs and of Aitchison coordinates. These are the building blocks which we use in Section 3 to construct the WA-GAN method.

2.1 A brief literature survey

For the terminology used in this survey see Section 2.2 below and see monographs on extreme value theory such as [7], [11] and [37].

In [2] a one-layer Re-LU network is combined with a new parametrization of a GAN to generate multivariate extreme data from heavy-tailed distributions. The method is called EV-GAN and is shown to be superior to a standard GAN.

ExtVAE [27] is based on the assumption that the observations are multivariate regularly varying and that the latent variable in the variational autoencoder (VAE) follows an inverse Gamma distribution and, applying some further constraints to insure regular variation, simulates values of the radial component. It then uses another VAE to produce simulated angles on the unit L_1-sphere using a VAE with Dirichlet distributed latent variables. Simulated observations are then obtained by multiplying the radial variable with the the angular value.

Under the assumption that estimates of the true marginal tail indices are are available, HTGAN [18] transforms multivariate observations to the unit-Pareto scale, trains a GAN with heavy-tailed latent variables and transforms simulated variables back to original scale. It generates samples from the entire data set, not just from the extreme part.

GP-GAN [28] adaptively selects exceedance levels, fits univariate GP distributions to the excesses of these levels, uses a standard GAN to propose a spectral value on the exponential scale, and then uses the fitted univariate GP distributions and the spectral value to generate a multidimensional value on the original scale. This value is then sent to the discriminator for optimization of the spectral GAN.

The evtGAN [8] is aimed at multivariate block maxima observations. It uses one-dimensional Generalized Extreme Value (GEV) distributions to transform marginals to approximately uniform $[0, 1]$ distributions, then trains a convolutional GAN on the uniform scale, and finally applies the inverse transform of the marginal distribution to obtain simulated values on the scale of the observations. The structure of a multivariate GEV distribution is not used in the simulation of the uniform variables.

The paper [21] uses the GEV model to transform observations of block maxima to the Pickands dependence function scale and then uses either the spectral measure or the Pickands measure restrictions to train a GAN to simulate from the dependence function. The simulated observations are then transformed back to original scale. The authors argue that this approach also is useful for efficient simulation from parametric models.

Finally, DeepGauge [24] applies a different framework than the references above derived from the theory of geometric extremes. It uses a multi-layer perceptron with ReLu activation function to first model radial thresholds and then another multi-layer perceptron to find a gauge function which describes the boundary of the deterministic limiting shape of scaled sample clouds.

2.2 Regular variation and extreme value analysis

Multivariate regular variation and angular measure. Let $\boldsymbol{X} = (X_1, \ldots, X_d)$ be the random vector of interest with values in $\mathbb{R}^d$. Based on a random sample of the (unknown) distribution of $\boldsymbol{X}$, we wish to generate new samples with the same tail behavior as $\boldsymbol{X}$, even at levels beyond those encountered in the sample. Such extrapolation is possible only if we are willing to make some regularity assumptions. Our core assumptions to obtain accurate modeling of the tail of $\boldsymbol{X}$ are founded in the theory of multivariate regular variation [36]. There are two approaches to use this hypothesis.

1. Either we postulate it on the random vector $\boldsymbol{X}$ itself. This implies that all univariate tail indices are same.

2. Or we postulate it on a transformed vector $\boldsymbol{V} = v(\boldsymbol{X})$, where v is a coordinatewise transformation of $\boldsymbol{X}$ so that the d components of $\boldsymbol{V}$ all have the same distribution. This implies that the multivariate assumption only depends on the dependence structure of $\boldsymbol{X}$ and not on the margins, which can be assessed separately.

Here we choose the second approach and transform to unit-Pareto margins using the transformation

$$v : \boldsymbol{x} = (x_j)_{j=1}^d \mapsto v(\boldsymbol{x}) \stackrel{\text{def}}{=} \left(\frac{1}{1 - F_j(x_j)} \right)_{j=1}^d,$$

where F_j denotes the distribution function of X_j. We assume throughout that these marginal distribution functions are continuous. It is straightforward to see that after this transformation the margins of $\boldsymbol{V} = v(\boldsymbol{X})$ have unit-Pareto distributions, $\mathsf{P}(V_j > y) = 1/y$ for $y \geq 1$. Our first assumption describes the asymptotic dependence structure of V.

Assumption 1 (Multivariate regular variation)**.** *There exists a non-zero Borel measure ν on $\mathbb{E} \stackrel{\text{def}}{=} [0, \infty)^d \setminus \{\mathbf{0}\}$ which is finite on Borel sets bounded away from $\mathbf{0}$ and such that*

$$\lim_{t \to \infty} t \, \mathsf{P}\left(t^{-1}\boldsymbol{V} \in B\right) = \nu(B)$$

for all Borel sets B of $\mathbb{E}$ bounded away from $\mathbf{0}$ with $\nu(\partial B) = 0$, with ∂B the topological boundary of B.

Intuitively, the measure ν helps in describing the tail behavior of $\boldsymbol{V}$ as for large $t > 0$, we have $\mathsf{P}(\boldsymbol{V} \in tB) \approx \nu(B)/t$ where $tB \stackrel{\text{def}}{=} \{t\boldsymbol{x} : \boldsymbol{x} \in B\}$. The measure ν is often referred to as the *exponent measure* of $\boldsymbol{V}$, because of its appearance in the exponent of the expression of the multivariate extreme value distribution to which the law of $\boldsymbol{V}$ is attracted [11, Definition 6.1.7].

Our procedure below will benefit from an equivalent formulation of Assumption 1 in polar coordinates with respect to the L_1-norm $|\boldsymbol{x}|_1 \stackrel{\text{def}}{=} \sum_{j=1}^{d} |x_j|$ for $\boldsymbol{x} \in \mathbb{R}^d$. More concretely, it can be shown [37, Theorem 6.1] that due to a convenient homogeneity property of ν [11, Theorem 6.1.9], if we put

$$R \stackrel{\text{def}}{=} |\boldsymbol{V}|_1 \qquad \text{and} \qquad \boldsymbol{W} \stackrel{\text{def}}{=} R^{-1}\boldsymbol{V}, \tag{1}$$

where $\boldsymbol{W}$ takes values in the unit simplex $\Delta_{d-1} \stackrel{\text{def}}{=} \{\boldsymbol{x} \in [0,1]^d : |\boldsymbol{x}|_1 = 1\}$, there exists a probability measure Φ on Δ_{d-1} such that

$$\lim_{t\to\infty} \mathsf{P}\left(\boldsymbol{W} \in A \mid R \geq t\right) = \Phi(A) \tag{2}$$

for any Borel set $A \subseteq \Delta_{d-1}$ such that $\Phi(\partial A) = 0$.

The measure Φ is often referred to as the *angular (probability) measure* of $\boldsymbol{V}$ as it describes how its "angular" component $\boldsymbol{W}$ behaves when its radial part R becomes large. The set of possible probability measures that can be obtained in such a way forms a nonparametric class as the only constraint they should satisfy—due to the unit-Pareto margins of $\boldsymbol{V}$—are

$$\int_{\Delta_{d-1}} w_j \, \mathrm{d}\Phi(\boldsymbol{w}) = \frac{1}{d}, \qquad j \in \{1, \ldots, d\}, \tag{3}$$

wich in turn implies

$$\mathsf{P}(R > t) \sim \frac{d}{t}, \qquad \text{as } t \to \infty. \tag{4}$$

It is possible for Φ to have positive mass on one of the faces of the simplex Δ_{d-1}, that is, sets for which at least one of the coordinates is zero. In terms of extremes, it corresponds to scenarios where some components of $\boldsymbol{X}$ are large while others are small. These are called *extreme directions* and have to be treated by specific methods which our methodology is not able to handle, see, e.g., [31]. We exclude this situation here.

Assumption 2. *The measure Φ in (2) is concentrated on the open simplex $\Delta_{d-1}^{\circ} \stackrel{\text{def}}{=} \{\boldsymbol{x} \in (0,1)^d : |\boldsymbol{x}|_1 = 1\}$, that is, if $\boldsymbol{\Theta} \sim \Phi$, then*

$$\mathsf{P}\left(\boldsymbol{\Theta} \in \Delta_{d-1}^{\circ}\right) = 1.$$

Univariate generalized Pareto distributions. Assumption 1 is not sufficient by itself to model the tail behavior of $\boldsymbol{X}$ as it only concerns the dependence structure. Our second main assumption concerns the tails of the margins X_j and allows for heterogeneous tails. Here and below, u_{*j} denotes the (possibly infinite) upper endpoint of X_j.

Assumption 3 (Generalized Pareto limit of marginal tails)**.** *For every $j \in \{1, \dots, d\}$, there exist $\xi_j \in \mathbb{R}$ and a function $\sigma_j(\cdot) : (-\infty, u_{*j}) \to (0, \infty)$ such that for all $y > 0$,*

$$\lim_{u \nearrow u_{*j}} \mathsf{P}\left(\frac{X_j - u}{\sigma_j(u)} > y \,\middle|\, X_j > u \right) = (1 + \xi_j y)_+^{-1/\xi_j} ,$$

where $a_+ \stackrel{\text{def}}{=} \max(a, 0)$ for $a \in \mathbb{R}$; if $\xi_j = 0$, the limit is to be understood as $\exp(-y)$.

By the Pickands–Balkema–de Haan theorem [6, 34], Assumption 3 is equivalent to the assumption that X_j is in the maximum domain of attraction of a GEV distribution. The limit in Assumption 3 holds uniformly in $y > 0$. The assumption justifies the following approximation for $y > 0$ such that $1 + \xi_j y/\sigma_j > 0$:

$$\mathsf{P}\left(X_j - u > y \,\middle|\, X_j > u \right) \approx \left(1 + \frac{\xi_j y}{\sigma_j} \right)^{-1/\xi_j} , \tag{5}$$

where $\sigma_j > 0$ is a scale parameter that accounts for the unknown function $\sigma_j(\cdot)$. Estimation of the bivariate parameter (σ_j, ξ_j) is typically performed by the method of maximum likelihood. This is the well-known *Peaks-Over-Threshold (PoT)* approach for modeling the tails of real-valued data, see for instance [42] and [7, Section 5.3]. A random variable with distribution function

$$H_{\sigma,\xi}(y) \stackrel{\text{def}}{=} 1 - \left(1 + \frac{\xi y}{\sigma} \right)_+^{-1/\xi} , \qquad y > 0,$$

for some $(\sigma, \xi) \in (0, \infty) \times \mathbb{R}$ is said to have a *generalized Pareto (GP) distribution.*

Remark 1 (Equivalent formulations of Assumption 3). As already explained, Assumption 3 is equivalent to the hypothesis that X_j is in the maximum domain of attraction of a GEV distribution. As a consequence, X_j satisfies all the equivalent formulations of this condition as presented, e.g., in Theorem 1.1.6 of [11]. As some of them will be needed below, we recall them here. Let $b_j(t) \stackrel{\text{def}}{=} F_j^{-1}(1 - 1/t) = (1/(1 - F_j))^{-1}(t)$, for $t > 1$, be the tail quantile function of F_j. Then, the following statements are equivalent to Assumption 3.

(i) There exist $\xi_j \in \mathbb{R}$ and a scaling function $a_j(\cdot) : (1, \infty) \to (0, \infty)$ such that

$$\lim_{t \to \infty} \mathsf{P}\left(\frac{X_j - b_j(t)}{a_j(t)} > y \mid X_j > b_j(t) \right) = (1 + \xi_j y)_+^{-1/\xi_j} , \qquad y > 0.$$

(ii) There exist $\xi_j \in \mathbb{R}$ and a scaling function $a_j(\cdot) : (1, \infty) \to (0, \infty)$ such that

$$\lim_{t \to \infty} \frac{b_j(ty) - b_j(t)}{a_j(t)} = \frac{y^{\xi_j} - 1}{\xi_j}, \qquad y > 0,$$

where the right-hand side is to be interpreted as $\log y$ if $\xi_j = 0$. The convergence also holds locally uniformly in $y \in (0, \infty)$, see Section 1.2 in [11].

The scalar $\xi_j \in \mathbb{R}$ in (i) and (ii) is equal to the one in Assumption 3, while the scaling function $a_j(\cdot)$ in (i) and (ii) is related to $\sigma_j(\cdot)$ in Assumption 3 via $\sigma_j(u) = a_j(1/(1-F_j(u)))$ for $u < u_{*j}$.

Multivariate generalized Pareto distributions. What do Assumptions 1–3 tell us about the tail behavior of $\boldsymbol{X}$? The answer can be conveniently formulated in terms of multivariate generalized Pareto (MGP) distributions, which extend Assumption 3 to the multivariate case. These distributions have been introduced in [40] and are reviewed in [32]. Below, we provide an introduction to their usage in modeling multivariate extremes along with a theoretical result (Proposition 1) which will be useful for sampling from a MGP distribution.

In arbitrary dimension, it is not clear how to define an "extreme" point. Some authors are interested in modeling the vector of componentwise maxima of the data, leading to the multivariate GEV distributions, studied extensively in the literature after the pioneering work of [12]. Recently, see e.g. [39], more attention has been given to an extension of the standard PoT procedure in a multivariate context. The approach relies on the fact that the excess of $\boldsymbol{X}$ above a large threshold vector $\boldsymbol{u}$ asymptotically follows an MGP distribution, with an appropriate definition of when an exceedance over a multivariate threshold takes place.

More concretely, given a vector $\boldsymbol{u} \in \mathbb{R}^d$ of "high thresholds", i.e., below but close to $\boldsymbol{u}_* = (u_{*j})_{j=1}^d$, we consider the random vector of excesses

$$\boldsymbol{Y}_{\boldsymbol{u}} \stackrel{\text{def}}{=} \boldsymbol{X} - \boldsymbol{u} \mid \boldsymbol{X} \not\leq \boldsymbol{u}, \tag{6}$$

where $\boldsymbol{X} - \boldsymbol{u} \stackrel{\text{def}}{=} (X_j - u_j)_{j=1}^d$ and where $\not\leq$ means that, for at least one $j \in \{1, \ldots, d\}$, we have $X_j > u_j$. Here and below, operations on real numbers are extended to vectors in $\mathbb{R}^d$ in a coordinate-wise manner. Under the aforementioned assumptions, we can establish the asymptotic distribution of $\boldsymbol{Y}_{\boldsymbol{u}}$ as $\boldsymbol{u} \to \boldsymbol{u}_*$, in a particular way made precise in the statement of the result. Weak convergence in $\mathbb{R}^d$ is denoted by $\stackrel{\mathcal{L}}{\to}$.

Proposition 1 (Weak convergence of excesses to the MGP distribution)**.** *Under Assumptions 1–3, we have*

$$\frac{\boldsymbol{X} - \boldsymbol{b}(t)}{\boldsymbol{a}(t)} \,\Big|\, \boldsymbol{X} \not\leq \boldsymbol{b}(t) \stackrel{\mathcal{L}}{\to} \frac{\boldsymbol{Y}^{\boldsymbol{\xi}} - 1}{\boldsymbol{\xi}}, \qquad t \to \infty, \tag{7}$$

where $\boldsymbol{Y}$ *is a* multivariate Pareto *random vector, with distribution*

$$\boldsymbol{Y} \stackrel{\mathcal{L}}{=} (Y\boldsymbol{\Theta} \mid Y\boldsymbol{\Theta} \not\leq 1) \tag{8}$$

with Y *a unit-Pareto random variable independent of* $\boldsymbol{\Theta} \sim \Phi$*, and where* $\boldsymbol{\xi} = (\xi_j)_{j=1}^d$ *is the vector of marginal coefficients as in Assumption 3;* $\boldsymbol{a}(\cdot) = (a_j(\cdot))_{j=1}^d$ *and* $\boldsymbol{b}(\cdot) = (b_j(\cdot))_{j=1}^d$ *are the same functions as the one introduced in Remark 1. Consequently, along the curve* $\boldsymbol{u} : t \in (1, \infty) \mapsto \boldsymbol{u}(t) \stackrel{\text{def}}{=} \boldsymbol{b}(t)$ *we have,*

$$\frac{\boldsymbol{Y}_{\boldsymbol{u}(t)}}{\boldsymbol{\sigma}(\boldsymbol{u}(t))} \stackrel{\mathcal{L}}{\to} \frac{\boldsymbol{Y}^{\boldsymbol{\xi}} - 1}{\boldsymbol{\xi}}, \qquad t \to \infty, \tag{9}$$

where $\boldsymbol{\sigma}(\cdot) = (\sigma_j(\cdot))_{j=1}^d$ *contains the scaling functions in Assumption 3.*

The proof of Proposition 1 is provided in Appendix A.

Typically, as for the univariate case, the scaling function in (9) is viewed as a scaling parameter and Proposition 1 leads to the approximation in distribution for

$$\boldsymbol{Y}_{\boldsymbol{u}(t)} \overset{\mathcal{L}}{\approx} \boldsymbol{H} \overset{\text{def}}{=} \boldsymbol{\sigma} \frac{\boldsymbol{Y}^{\boldsymbol{\xi}} - 1}{\boldsymbol{\xi}}, \qquad t \to \infty \tag{10}$$

with $\boldsymbol{\sigma} \in (0, \infty)^d$ and $\boldsymbol{\xi} \in \mathbb{R}^d$ containing, respectively, the scale and shape parameters from the marginal GP distributions in Assumption 3 and where $\boldsymbol{Y}$ is Multivariate Pareto distributed as in (8).

Remark 2 (Standard MGP random vectors). The random vector $\boldsymbol{H}$ in (10) is $\text{MGP}(\boldsymbol{\sigma}, \boldsymbol{\xi}, \boldsymbol{S})$ distributed, in the sense of [32, Definition 2.2], with *spectral generator* $\boldsymbol{S}$ having distribution

$$\mathsf{P}\left(\boldsymbol{S} \leq \boldsymbol{x}\right) = \frac{\mathsf{E}\left[\exp(Q)\, \mathbb{I}\{\boldsymbol{U} \leq \boldsymbol{x} + Q\}\right]}{\mathsf{E}[\exp(Q)]}, \qquad \boldsymbol{x} \in \mathbb{R}^d, \tag{11}$$

where $\boldsymbol{U} \overset{\text{def}}{=} \log(\boldsymbol{\Theta})$, with $\boldsymbol{\Theta} \sim \Phi$, and $Q \overset{\text{def}}{=} \max(\boldsymbol{U})$. In other words, the random vector $\boldsymbol{U}$ is a U-generator of $\boldsymbol{H}$ in the sense of Proposition 9 of [38]; see also equation (12) in [32]. Note in particular that the moment condition $0 < \mathsf{E}[\exp(U_j)] < \infty$ for all $j \in \{1, \ldots, d\}$ is satisfied as we have, from (3),

$$\mathsf{E}[\exp(U_j)] = \mathsf{E}[\Theta_j] = \int_{\Delta_{d-1}} w_j \, \mathrm{d}\Phi(\boldsymbol{w}) = \frac{1}{d}, \qquad j \in \{1, \ldots, d\}.$$

Here, since we work mainly with the angular measure Φ in our procedure, we decided to consider the multivariate Pareto random vector $\boldsymbol{Y}$, as in (8), to be the "standard" MGP, that is with $\boldsymbol{\xi} = \mathbf{1}$ and $\boldsymbol{\sigma} = \mathbf{1}$ instead of the choice $\boldsymbol{\xi} = \mathbf{0}$ made in [38, 32, 39], for example. This approach has also been considered in [17]. Of course, it suffices to take $\boldsymbol{Z} = \log(\boldsymbol{Y})$ to obtain an MGP random vector with parameter $\boldsymbol{\xi} = \mathbf{0}$, as can be seen from comparing (10) to Equation (4) in [32].

Tail modeling and rare event probabilities. Thanks to Proposition 1, we have now a clear path to modeling the tail of $\boldsymbol{X}$. Suppose we are interested in estimating a probability $\mathsf{P}(\boldsymbol{X} \in C)$ for some $C \subseteq \{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{x} \not\leq \boldsymbol{u}(t)\}$ for some large $t > 1$, with the same $\boldsymbol{u}(t)$ as in (9), so that the approximation of $\boldsymbol{Y}_{\boldsymbol{u}(t)} = \boldsymbol{X} - \boldsymbol{u}(t) \mid \boldsymbol{X} \not\leq \boldsymbol{u}(t)$ by the MGP $\boldsymbol{H}$ as in (10) is accurate. In such a region, there may be no or few observations available, so that an empirical estimate is not reliable. Instead, we write the probability of interest as

$$\begin{aligned} \mathsf{P}(\boldsymbol{X} \in C) &= \mathsf{P}(\boldsymbol{X} \not\leq \boldsymbol{u}(t))\ \mathsf{P}\left(\boldsymbol{Y}_{\boldsymbol{u}(t)} \in C - \boldsymbol{u}(t)\right) \\ &\approx \mathsf{P}(\boldsymbol{X} \not\leq \boldsymbol{u}(t))\ \mathsf{P}\left(\boldsymbol{H} \in C - \boldsymbol{u}(t)\right). \end{aligned} \tag{12}$$

The first probability can be estimated using an empirical evaluation on the sample while the second one will be computed using an estimate of the MGP distribution of $\boldsymbol{H}$.

As can be seen from (10), the MGP random vector $\boldsymbol{H}$ is a simple transformation of the standard MGP random vector $\boldsymbol{Y}$ introduced in (8), involving only marginal parameters for which estimation is well developed mathematically and numerically. Therefore, the main challenge to obtain informative samples to evaluate tail probabilities related to $\boldsymbol{X}$ is to obtain realizations of $\boldsymbol{Y}$. By Proposition 1, this is possible if we are able to simulate from Φ—at least approximately since it is a limit distribution by nature, so that no direct observation of it will ever be available.

We propose to use a specific GAN structure to achieve this goal. The main principles underlying this framework are recalled in the following section.

2.3 Wasserstein Generative Adversial Networks

GANs have been introduced by Ian Goodfellow and co-authors in [19]. Initially, they have been proposed for synthetic image generation but since then, they have been applied to any kind of data, in particular tabular data which are of particular interest to statisticians [4].

Suppose we are interested in estimating the unknown distribution P of some, potentially complex, data on a large vector space $\mathcal{X}$. For instance, think of the angular measure Φ in Section 2.3 supported on the simplex Δ_{d-1} for large d. Then, the standard parametric approaches based on densities may not be satisfactory as they are often too restrictive to take into account all the possible characteristics of a complex angular measure in large dimension. An alternative direction is to start from a simple random variable $Z \in \mathcal{Z}$ in some "latent" space–typically, of lower dimension than $\mathcal{X}$–whose distribution is known and to look for a parametric transformation $g_\theta : \mathcal{Z} \to \mathcal{X}$ such that $g_\theta(Z)$ has a distribution sufficiently close to P. Practically, this has several advantages. We get more flexibility as the map g_θ can be virtually anything if we consider a sufficiently rich family of parametric functions and the latent space/distribution can also be selected freely. It is easier to simulate from P as it suffices to obtain a sample of Z (easy) and transform it based on g_θ rather than more costly algorithms based on the knowledge of the density values.

A Wasserstein GAN (WGAN) architecture [5] follows this path to estimate P by considering g_θ to be a neural network and by measuring the difference in distribution between the real data distribution P and the law of $g_\theta(Z)$ via the *Earth Mover's Distance (EMD)*, which is a particular instance of the Wasserstein distance in optimal transport [44].

A WGAN consists of two neural networks: a so-called "Generator" $G = G_\theta : \mathcal{Z} \mapsto \mathcal{X}$ and a "Discriminator" $D = D_w : \mathcal{X} \mapsto \mathbb{R}$ (also sometimes called "Critic" depending on the authors). These two networks play an opposite role: G_θ aims to produce samples as realistic as possible and close to the target distribution and D_w aims at efficiently discriminating samples and determining which ones are generated by G_θ and which ones are real. This "game" can be formulated as a min-max optimization problem in the EMD setup that we recall know.

The EMD is simply a distance between probability measures with finite first moment defined on a metric space $(\mathcal{X}, \rho)$. If P and Q are two such distributions, it is equivalently

defined as

$$W_1(\mathsf{P},\mathsf{Q}) \stackrel{\text{def}}{=} \inf_{\pi\in\Pi(\mathsf{P},\mathsf{Q})} \int_{\mathcal{X}\times\mathcal{X}} \rho(x,y)\,\mathrm{d}\pi(x,y) \tag{13}$$

$$= \sup_{f\in \mathrm{Lip}_1} \left\{ \int_{\mathcal{X}} f(x)\,\mathrm{d}\mathsf{P}(x) - \int_{\mathcal{X}} f(x)\,\mathrm{d}\mathsf{Q}(x) \right\}, \tag{14}$$

where $\Pi(\mathsf{P},\mathsf{Q})$ is the set of probability measures on $\mathcal{X}\times\mathcal{X}$ with P and Q as first and second margins and Lip_1 is the set of functions $f:\mathcal{X}\to\mathbb{R}$ that are 1-Lipschitz continuous, that is, functions that satisfy

$$\sup_{x,y\in\mathcal{X},x\neq y} \frac{|f(x)-f(y)|}{\rho(x,y)} \leq 1.$$

If $\mathcal{X}$ is an open subset of $\mathbb{R}^N$ for some $N\in\mathbb{N}$, this implies

$$|\nabla f(x)|_2 \leq 1 \qquad \text{for almost every } x\in\mathcal{X}, \tag{15}$$

that is, for every $x\in\mathcal{X}$ except for a subset of Lebesgue measure zero. The equivalence between (13) and (14) is known as the Kantorovich–Rubinstein duality theorem, the proof of which is to be found in, e.g., [44, Section 1.2].

In formulation (14) of the EMD, the discriminator D_w will play the role of the "separating" function f and Q will correspond to the measure generated by the generator G_θ, that is, the distribution of $G_\theta(Z)$, while P will again be our target measure from which we observe a random sample. This leads to the following min-max procedure:

$$\min_{\theta\in\mathcal{S}_G}\max_{w\in\mathcal{S}_D} \left\{ \int_{\mathcal{X}} D_w(x)\,\mathrm{d}\mathsf{P}(x) - \int_{\mathcal{Z}} D_w\left(G_\theta(z)\right)\,\mathrm{d}\mathsf{P}_Z(z) \right\}, \tag{16}$$

where P_Z denotes the law of the latent vector Z and $\mathcal{S}_G$ and $\mathcal{S}_D$ denote the (finite-dimensional) parameter spaces of the generator and the discriminator respectively. Optimization is performed in the respective parameter spaces of the generator and the discriminator, which contain all the weights in their respective network structures. Typically, the expectations are replaced by averages over batches of real data with distribution P and fake data with distribution P_Z. The joint minimization/maximization of the objective function is performed by gradient descent based on the standard backpropagation algorithm. The whole GAN procedure is summarized in Figure 1.

In practice there is no need for the neural network D_w to be Lipschitz continuous as required in definition (14) of the EMD. Initially, in [5], authors proposed to clip the weights, that is, forcing them to lie in a fixed compact space, typically $[-c,c]^{|\mathcal{S}_D|}$ for some small $c>0$. One may show that if the discriminator consists of a multilayer perceptron, which will always be the case here, its Lipschitz norm will be bounded by a finite constant depending only on c. Even though this constant may not be equal to one, replacing Lip_1 by Lip_C for $C>0$ in (14) only amounts to multiplying the value of the resulting distance by $C>0$ and it does not play a role in the optimization of the generator.

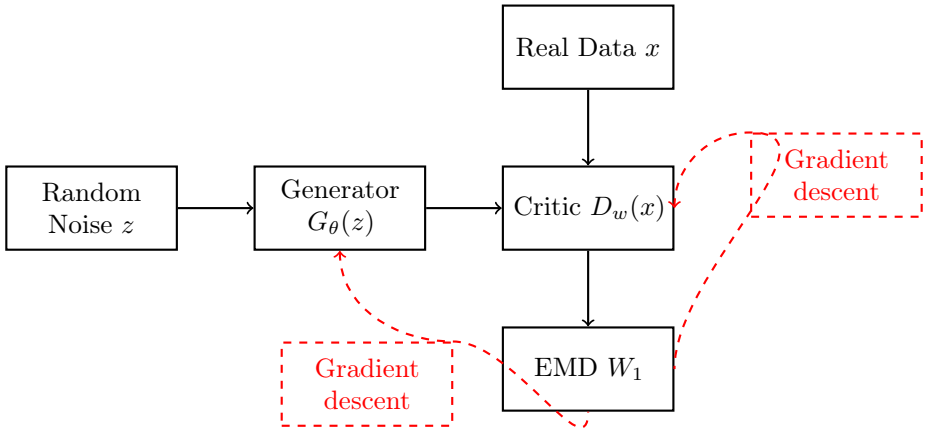


Figure 1: Illustration of a Wasserstein GAN with gradient-based optimization

Since then, it has been shown that in some cases it is preferable to enforce the Lipschitz constraint in a soft manner, using (15). Of course, controlling the gradient ∇D_w everywhere on $\mathcal{X}$ is not possible, so what is usually done is to add a penalty term to the objective function of the form

$$\lambda \int_{\mathcal{X}} (|\nabla D_w(x)|_2 - 1)^2 \, \mathrm{d}\mu(x), \qquad \lambda > 0, \tag{17}$$

where μ is a probability measure on $\mathcal{X}$. The standard approach in [20] is to take μ as the law of the mixture $U \cdot X_r + (1-U) \cdot G_\theta(Z)$ where $X_r \sim \mathsf{P}$ follows the data distribution, $Z \sim \mathsf{P}_Z$ is random noise and U is uniformly distributed on $[0,1]$ and independent of the rest. This choice is motivated by Proposition 1 in [20] and we follow this approach here.

As explained at the end of Section 2.2 and in the beginning of this section, our aim is to use this architecture to obtain accurate samples from the angular measure Φ on Δ_{d-1}. Since the simplex structure of such observations makes their distribution a bit non-standard and could prevent the generator from producing high-quality output, we propose to apply the procedure on a transformation of the angular data to coordinates in an orthonormal basis of the simplex and to transform the coordinates back to angles afterwards. The next section and Appendix B provide more details about this transformation.

2.4 Aitchison coordinates

A core idea of compositional data analysis [1, 13], mainly developed by John Aitchison (1926–2016), consists of transforming raw simplex data in $\mathbb{R}^d$ to coordinates with respect to an orthonormal basis of a certain $(d-1)$-dimensional subspace of $\mathbb{R}^d$. This makes interpretation more complicated, but having data on a full linear $(d-1)$-dimensional sample space instead of on the unit simplex in $\mathbb{R}^d$ can improve statistical and neural network modeling substantially. Since we are not all that concerned with interpretation

in the WGAN architecture described in Section 2.3 but instead with obtaining the highest possible efficiency in producing accurate new multivariate extreme samples, we will apply this idea in our context.

Infinitely many orthogonal bases exist for the open simplex $\Delta_{d-1}^{\mathrm{o}}$ equipped with its inner product space structure. Here, we work with the one proposed in Proposition 2 of [13]. Details of the underlying construction and more information on the Aitchison simplex, i.e., the unit simplex equipped with a certain vector space structure and an inner product, can be found in Appendix B. The essence of the matter is to map $\Delta_{d-1}^{\mathrm{o}}$ to $\mathbb{H} = \{\boldsymbol{x} \in \mathbb{R}^d : \sum_{j=1}^d x_j = 0\}$ by means of the centered logratio function (clr) in (28). The set $\mathbb{H}$ is a $(d-1)$-dimensional linear space equipped with the standard Euclidean inner product, the structure of which carries over to $\Delta_{d-1}^{\mathrm{o}}$ through the function clr.

Proposition 2 (An orthonormal basis for the Aitchison simplex)**.** *The vectors* $\boldsymbol{e}_1^*, \ldots, \boldsymbol{e}_{d-1}^*$ *defined by*

$$\boldsymbol{e}_i^* \stackrel{\text{def}}{=} \operatorname{softmax}(e_i) \in \Delta_{d-1}^{\mathrm{o}} \text{ with } e_i \stackrel{\text{def}}{=} \sqrt{\frac{i}{i+1}} \left(\underbrace{i^{-1}, \ldots, i^{-1}}_{i \text{ times}}, -1, 0 \ldots, 0 \right) \in \mathbb{R}^d$$

for $i \in \{1, \ldots, d-1\}$ *form an orthonormal basis of the Aitchison simplex.*

Now that we have constructed an orthonormal basis of the Aitchison simplex in Proposition 2, we are able to decompose any vector $\boldsymbol{u} \in \Delta_{d-1}^{\mathrm{o}}$ in this basis as

$$\boldsymbol{u} = \bigoplus_{i=1}^{d-1} \langle \boldsymbol{u}, \boldsymbol{e}_i^* \rangle_A \odot \boldsymbol{e}_i^* = \bigoplus_{i=1}^{d-1} \langle \operatorname{clr}(\boldsymbol{u}), \operatorname{clr}(\boldsymbol{e}_i^*) \rangle \odot \boldsymbol{e}_i^* = \bigoplus_{i=1}^{d-1} \langle \operatorname{clr}(\boldsymbol{u}), \boldsymbol{e}_i \rangle \odot \boldsymbol{e}_i^*.$$

the notation being defined in Appendix B. The coordinates $\boldsymbol{u}_i^* \stackrel{\text{def}}{=} \langle \operatorname{clr}(\boldsymbol{u}), \boldsymbol{e}_i \rangle$ for $i \in \{1, \ldots, d-1\}$ are sufficient to completely describe the vector $\boldsymbol{u}$. These coordinates will be the quantities entering the WGAN structure described in Section 2.3 as they typically have a nicer distribution than the original simplex random vectors of interest and hence will be easier to "learn", which helps facing the difficulty that sample sizes in extreme value analysis are often not very large.

3 Combining GAN, Aitchison coordinates and Extremes

Recall that our main objective is to provide an algorithm which is able to accurately sample from the tail of a random vector $\boldsymbol{X} \in \mathbb{R}^d$. We do this by combining Algorithm 1 and Algorithm 2 described below.

- Algorithm 1 develops a Wasserstein GAN to simulate from the dependence structure of the unit-Pareto transformed observations.
- Algorithm 2 uses estimates of the marginal GPD parameters to transform these simulated values to the original scale.

The simulated values can then be used to obtain accurate estimates of probabilities of the form $\mathsf{P}(\boldsymbol{X} \in C)$ where $C \subseteq \{\boldsymbol{x} \in \mathbb{E} : \boldsymbol{x} \not\leq \boldsymbol{u}(t)\}$ for some "high" threshold vector $\boldsymbol{u}(t) \in \mathbb{R}^d$ as in (9), close to $\boldsymbol{u}_*$.

The main challenge lies in simulation from the angular measure Φ of $\boldsymbol{V}$ in Assumption 1, since moving from Φ to the MGP distribution is a matter of scaling and marginal tail estimation, for which theory and practice are well developed. Consequently, we propose to use the WGAN architecture of Section 2.3 to obtain accurate samples from Φ. This is done via the Aitchison coordinates, Section 2.4, as detailed in Algorithm 1 below.

The input to Algorithm 1 is a set of training observations $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$, with $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{id})$, which will be empirically converted to their unit-Pareto margins version using the empirical marginal cumulative distribution functions

$$\widehat{F}_j(x) \stackrel{\text{def}}{=} \frac{1}{n+1} \sum_{i=1}^{n} \mathbb{I}\{X_{ij} \leq x\}, \qquad x \in \mathbb{R}, j \in \{1, \ldots, d\}. \tag{18}$$

Division by $n+1$ instead of n is there to prevent division by zero in the standardization. The angles associated to "large" observations are supposed to provide an approximate sample from Φ in view of Assumption 1. The output of Algorithm 1 is a trained generator G_θ which is able to produce new coordinates in the Aitchison orthonormal basis, defined by the orthonormal basis $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{d-1}\}$ of $\mathbb{H}$, that are hopefully close in distribution to the coordinates of angles associated to the large observations in the training set.

The WA-GAN procedure in Algorithm 1 only uses the observations for which $R = |\boldsymbol{V}|_1$ is larger than $t = n/k_1$ so that, by (4), the number of observations used by the algorithm is asymptotically binomial with parameters n and $(dk_1)/n$. Here k_1 is a value to be chosen by the user, with larger k_1 meaning that more observations are used and hence the variance is smaller, and with smaller k_1 (hopefully) making the approximation in Assumption 1 more accurate. Finite sample bounds on the quality of this approximation are given in [9]. The relatively low effective sample size makes it important that the architectures of the generator and the discriminator are simple enough so that their parameters can be learned from a moderate number of examples. Those considerations are further explored in the numerical sections.

Algorithm 2 uses for each margin the k_2 largest observations, so that between k_2 and dk_2 of the multivariate observations are used. According to Remark 4, the value of k_2 should not be larger than k_1. A natural convenient choice is to take $k_1 = k_2$.

Algorithm 1 is essentially the standard WGAN algorithm with gradient penalty of [20] applied to the coordinates of the large angles in an orthonormal basis of the Aitchison simplex. The ADAM optimizer which is used to update the weights of the generator and the discriminator at each step is a popular gradient descent algorithm with adaptive learning rate introduced in [26]. The function `Adam()` in the algorithm is one such gradient step and is the default optimizer proposed for the WGAN architecture with gradient penalty. Compared to the standard algorithm, one additional regularization term

$$\rho \left| \frac{1}{m} \sum_{i=1}^{m} \operatorname{softmax}\left(V G_\theta(\boldsymbol{z}_i)\right) - \frac{\mathbf{1}_d}{d} \right|_2 \tag{19}$$

Algorithm 1 WA-GAN for tail dependence estimation

Require: observations $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$, orthonormal basis $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{d-1}\}$ of $\mathbb{H}$, and $k_1 \in \{1, \ldots, n\}$

Require: gradient penalty coefficient $\lambda > 0$, marginal penalty coefficient $\rho > 0$, the number of discriminator iterations per generator iteration n_D, the batch size m, Adam hyperparameters $\alpha > 0$ (learning rate), $\beta_1, \beta_2 \in [0, 1)$, the latent space dimension $\ell \in \mathbb{N}$, the number of epochs n_E

Require: discriminator initial parameters $w_0 \in \mathcal{S}_D$, generator initial parameters $\theta_0 \in \mathcal{S}_G$

$t \leftarrow n/k_1$

for $i = 1, \ldots, n$ **do**

$\widehat{\boldsymbol{V}}_i \leftarrow \left(1/(1 - \widehat{F}_j(X_{ij})\right)_{j=1}^{d}$

$R_i \leftarrow |\widehat{\boldsymbol{V}}_i|_1$ and $\boldsymbol{W}_i \leftarrow \widehat{\boldsymbol{V}}_i / R_i$

if $R_i \geq t$ **then**

$\boldsymbol{W}_i^* \leftarrow \langle \mathrm{clr}(\boldsymbol{W}_i), \boldsymbol{e}_i \rangle$

end if

end for

$K \leftarrow \sum_{i=1}^{n} \mathbb{I}\{R_i \geq t\}$

$V \leftarrow (\boldsymbol{e}_1 \,|\, \ldots \,|\, \boldsymbol{e}_{d-1}) \in \mathbb{R}^{d \times (d-1)}$

$\theta \leftarrow \theta_0$ and $w \leftarrow w_0$

for epoch $e = 1, \ldots, n_E$ **do**

for $t = 1, \ldots, n_D$ **do**

for $i = 1, \ldots, m$ **do**

sample $\boldsymbol{w}_i^*$ from $\{\boldsymbol{W}_1^*, \ldots, \boldsymbol{W}_K^*\}$, sample $\boldsymbol{z}_i$ from a standard multivariate normal distribution on $\mathbb{R}^\ell$, sample u_i from an uniform distribution on $[0, 1]$

$\widetilde{\boldsymbol{w}}_i^* \leftarrow G_\theta(\boldsymbol{z}_i)$

$\widehat{\boldsymbol{w}}_i^* \leftarrow u_i \, \boldsymbol{w}_i^* + (1 - u_i) \, \widetilde{\boldsymbol{w}}_i^*$

end for

$L_D \leftarrow \frac{1}{m} \sum_{i=1}^{m} \left\{ D_w(\widetilde{\boldsymbol{w}}_i^*) - D_w(\boldsymbol{w}_i^*) + \lambda \left(|\nabla_{\boldsymbol{w}^*} D_w(\widehat{\boldsymbol{w}}_i^*)|_2 - 1 \right)^2 \right\}$

$w \leftarrow \mathrm{Adam}(L_D, w, \alpha, \beta_1, \beta_2)$

end for

for $i = 1, \ldots, m$ **do**

sample $\boldsymbol{z}_i$ from a standard multivariate normal distribution on $\mathbb{R}^\ell$

end for

$L_G \leftarrow -\frac{1}{m} \sum_{i=1}^{m} D_w \left(G_\theta(\boldsymbol{z}_i) \right) + \rho \left| \frac{1}{m} \sum_{i=1}^{m} \mathrm{softmax}\left(V G_\theta(\boldsymbol{z}_i) \right) - \frac{\mathbf{1}_d}{d} \right|_2$

$\theta \leftarrow \mathrm{Adam}(L_G, \theta, \alpha, \beta_1, \beta_2)$

end for

Output: Trained generator G_θ

is added to enforce the marginal constraint (3) in the generator in a soft way and thus to enforce sampling from a genuine angular measure.

Algorithm 1 focuses solely on the extremal dependence of $\boldsymbol{X}$ and is independent of any marginal assumptions besides continuity of the marginal distribution functions. Therefore, if one is interested in computing a quantity which only depends on Φ in Assumption 1, such as the coefficients θ_J introduced in the beginning of Section 4, this can be done independently of Algorithm 2. This is not the case for methods such as [28] which directly evaluate sample quality on the scale of the MGP distribution. Note also that even though Algorithm 1 only considers the angular measure Φ with respect to the L_1-norm, a simple post-processing step allows for any norm on $\mathbb{R}^d$, as explained in Appendix C.

Remark 3 (Standardization of the margins to unit-Pareto). The procedure in Algorithm 1 is based on data standardized to unit-Pareto margins, in accordance with Assumption 1. As in (18), we work with an empirical standardization $\widehat{F}_j$ for each margin $j = 1, \ldots, d$, resulting in a procedure using only the marginal ranks of the original data and solely focusing on dependence, not requiring any assumption on the tail of F_j as in Assumption 3. The same approach was followed in [14, 16] and, more recently, in [9], for example. An alternative would be to use the empirical distribution function $\widehat{F}_j$ only up to a threshold $u_j < u_{*j}$ and fit a univariate GP distribution above this threshold, relying on Assumption 3. Doing so could improve the quality of the transformation to unit-Pareto margins, but at the price of making Algorithm 1 also require marginal assumptions. Furthermore, this route has already been explored in [15] and the same authors advocate in [14] for the pure nonparametric approach. Therefore, we stick to the rank-based procedure.

Some care needs is needed when sampling from $\boldsymbol{X} \mid \boldsymbol{X} \not\leq \boldsymbol{u}(t)$ for $t > 1$ large, where $\boldsymbol{u}(t)$ is the vector of thresholds such that for each component separately, the excess probability is $1/t$. For example, a common situation is the case where $\boldsymbol{X}$ represents some positive multivariate risks, that is, $\boldsymbol{X}$ takes values in $[0, \infty)^d$. If we directly make use of the formula $\boldsymbol{u}(t) + \boldsymbol{H}$, with estimated quantities, as suggested by formula (10), we risk generating "extreme" points for which some coordinates are negative. Our approach avoids this issue.

Algorithm 1 permits to sample from the multivariate Pareto distribution $\boldsymbol{Y}$ associated to $\boldsymbol{X}$ using (8). Let $\boldsymbol{Y}^*$ denote such a generated quantity. From (25), we have

$$\boldsymbol{Y}^* \overset{\mathcal{L}}{\approx} \left(\frac{k}{n} \boldsymbol{V} \mid \boldsymbol{V} \not\leq \frac{n}{k} \right),$$

where $\overset{\mathcal{L}}{\approx}$ means an approximation in distribution. Equivalently,

$$\boldsymbol{V}^* \overset{\text{def}}{=} \frac{n}{k} \boldsymbol{Y}^* \overset{\mathcal{L}}{\approx} \left(\boldsymbol{V} \mid \boldsymbol{V} \not\leq \frac{n}{k} \right).$$

Now, to pass from $\boldsymbol{V}$ to $\boldsymbol{X}$ one needs to apply the transformation $\boldsymbol{b}$ described in Proposition 1, which is based on the marginal quantile functions. Let $\widehat{\boldsymbol{b}}$ denote an estimator of this quantity. The final output of our next algorithm will be

$$\boldsymbol{X}^* = \widehat{\boldsymbol{b}}(\boldsymbol{V}^*) = \widehat{\boldsymbol{b}} \left(\frac{n}{k} \boldsymbol{Y}^* \right) \overset{\mathcal{L}}{\approx} \left(\boldsymbol{X} \mid \boldsymbol{V} \not\leq \frac{n}{k} \right) = \left(\boldsymbol{X} \mid \boldsymbol{X} \not\leq \boldsymbol{b} \left(\frac{n}{k} \right) \right),$$

by the continuity of margins. The estimator $\widehat{b}_j((n/k)y)$ for $y > 0$ depends on whether $y \leq 1$ or $y > 1$. For $y \leq 1$ we simply use the estimator based on the empirical cumulative distribution function $\widehat{F}_j$. For $y > 1$, the estimator is obtained through Assumption 3, more precisely its equivalent formulation (ii) in Remark 1. This leads to

$$\widehat{b}_j((n/k)y) \stackrel{\text{def}}{=} \begin{cases} X_{(\lceil n-k/y \rceil \vee 1):n,j}, & 0 < y \leq 1, \\ \widehat{b}_j(n/k) + \widehat{a}_j(n/k) \cdot \dfrac{y^{\widehat{\xi}_j} - 1}{\widehat{\xi}_j}, & y > 1, \end{cases} \tag{20}$$

where $X_{1:n,j} < \ldots < X_{n:n,j}$ are the ascending order statistics in the j-th sample, and where $\widehat{a}_j(n/k) \stackrel{\text{def}}{=} \widehat{\sigma}_j$ and $\widehat{\xi}_j$ are obtained by maximum likelihood for the GP distribution.

Algorithm 2 Sampling from the tail of $\boldsymbol{X}$

Require: observations $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$, orthonormal basis $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{d-1}\}$ of $\mathbb{H}$, trained generator G_θ, the size of the latent space on which G_θ has been trained ℓ, and $k_2 \in \{1, \ldots, n\}$
Require: desired number of samples n^*
  for $j = 1, \ldots, d$ **do**
    $u_j \leftarrow X_{n-k_2:n,j}$
    Compute the maximum likelihood estimator $(\widehat{\sigma}_j, \widehat{\xi}_j)$ of an univariate GPD
  end for
  $V \leftarrow (\boldsymbol{e}_1 \mid \ldots \mid \boldsymbol{e}_{d-1}) \in \mathbb{R}^{d \times (d-1)}$
  $i \leftarrow 1$ and $\mathcal{G}$ an empty list of size n^*
  while $i \leq n^*$ **do**
    sample $\boldsymbol{z}$ from a standard multivariate normal distribution on $\mathbb{R}^\ell$
    $\boldsymbol{W} \leftarrow \mathrm{clr}^{-1}(VG_\theta(\boldsymbol{z}))$
    sample Y from a unit-Pareto distribution
    $\boldsymbol{Y} \leftarrow Y\boldsymbol{W}$
    if $\max(\boldsymbol{Y}) > 1$ **then**
      for $j = 1, \ldots, d$ **do**
        if $Y_j \leq 1$ **then**
          $(\mathcal{G}[i])_j \leftarrow X_{(\lceil n-k_2/Y_j \rceil \vee 1):n,j}$
        end if
        if $Y_j > 1$ **then**
          $(\mathcal{G}[i])_j \leftarrow u_j + \widehat{\sigma}_j(Y_j^{\widehat{\xi}_j} - 1)/\widehat{\xi}_j$
        end if
      end for
      $i \leftarrow i + 1$
    end if
  end while
  Output: List of n^* random variables $\mathcal{G}$ approximately distributed as $\boldsymbol{X} \mid \boldsymbol{X} \not\leq \boldsymbol{u}$

Remark 4 (Relation between thresholds in both algorithms). When combining Algorithm 2 with the output of Algorithm 1, one actually uses two different thresholds: on the one

hand, the threshold $t_1 = n/k_1 > 0$ which has been used to fit the angular measure in Algorithm 1, and, on the other hand, the threshold vector $\boldsymbol{u}(t_2) = \boldsymbol{b}(n/k_2)$ for which we aim to sample from $\boldsymbol{X} \mid \boldsymbol{X} \not\leq \boldsymbol{u}(t_2)$. As Algorithm 2 relies on the output of Algorithm 1, one should ensure that $\boldsymbol{u}(t_2)$ is large enough so that the approximation of the angular measure Φ in Algorithm 1 is valid. In terms of the random variable R in (1), it means that we must ensure $R = |\boldsymbol{V}|_1 > n/k_1$ given that $\boldsymbol{X} \not\leq \boldsymbol{u}(t_2)$, which is equivalent to $\boldsymbol{V} \not\leq n/k_2$, i.e., $|\boldsymbol{V}|_\infty > n/k_2$. Since we have $|\boldsymbol{V}|_1 \geq |\boldsymbol{V}|_\infty$, it suffices to take $k_2 \leq k_1$. This is illustrated in Figure 2 for $d = 2$.

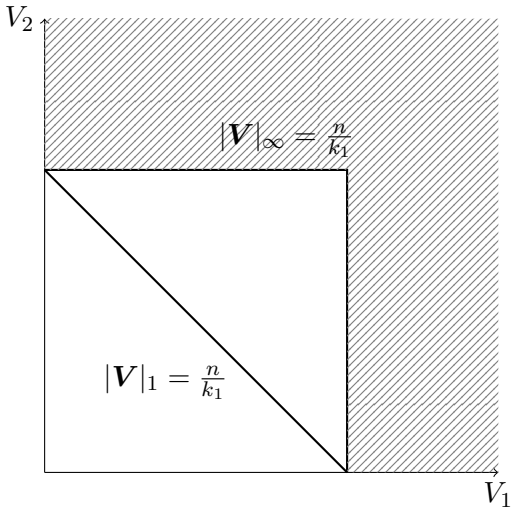


Figure 2: Relation between thresholds in the $\boldsymbol{V}$ space. The gray zone is where we can safely sample using Algorithm 2.

4 Numerical experiments

We use both simulated and real data sets to explore the performance of our proposed method and to compare it to other existing methods. We start with providing the details of how the different methods are compared and how the the computations are made. The next section uses simulated data to compare WA-GAN with HTGAN and GPGAN. In the final section these methods are applied to a financial data set.

4.1 Performance and computation

Performance measures. To assess the quality of the generated sample $\boldsymbol{X}^{\mathcal{G}}$, we will typically compare it with a sample from the data $\boldsymbol{X}^{\mathcal{T}}$ (test set) which has not been used for training. We propose to use measures that are inspired by multivariate extreme value theory since our focus is on the tail of $\boldsymbol{X}$.

A first set of measures that we propose is based on the *extremal coefficients* [7, Section 8.2.7]. For any non-empty subset $J \subseteq \{1, \ldots, d\}$ with at least two elements

$|J| \geq 2$, we define its extremal coefficient θ_J as

$$\theta_J \stackrel{\text{def}}{=} d \cdot \int_{\Delta_{d-1}} \bigvee_{j \in J} w_j \, \mathrm{d}\Phi(\boldsymbol{w}) \in [1, |J|]. \tag{21}$$

The closer θ_J is to 1, the stronger the dependence in the tail of $(X_j)_{j \in J}$, while θ_J close to $|J|$ corresponds to weaker tail dependence. For a given coefficient order $k = |J|$, to compute a summary score of their difference in the test set $\boldsymbol{X}^{\mathcal{T}}$ and the generated set $\boldsymbol{X}^{\mathcal{G}}$, we compute their relative absolute difference as follows:

$$E_{\overline{\theta}}(k) \stackrel{\text{def}}{=} \frac{1}{\binom{d}{k}} \sum_{\substack{J \subseteq \{1,\ldots,d\} \\ |J|=k}} \left| 1 - \frac{\theta_J^{\mathcal{G}}}{\theta_J^{\mathcal{T}}} \right|. \tag{22}$$

The coefficients are estimated for the test set using the empirical angular measure in (21) and for the generative methods using the sampled realizations of Φ. For the test set, the radial threshold n/k is the same as for the training set. Typically, we will consider the bivariate coefficients, with $k = 2$, as in [8], and the trivariate coefficients, with $k = 3$. This measure focuses on the extremal dependence structure only and aims to measure the quality of Algorithm 1.

A second measure that we propose, which also focuses on the tail of $\boldsymbol{X}$, aims at evaluating the output of Algorithm 2. Let $\boldsymbol{X}^{\mathcal{T}}$ denote a set of observations distributed like $\boldsymbol{X}$, which typically have not been used for training, and let $\boldsymbol{X}_{\boldsymbol{u}}^{\mathcal{G}}$ denote a set of observations generated by Algorithm 2 based on the threshold vector $\boldsymbol{u} = \boldsymbol{b}(n/k_n) \in \mathbb{R}^d$. Then if the MGP approximation is accurate enough, we expect $\boldsymbol{X}_{\boldsymbol{u}}^{\mathcal{T}} \stackrel{\text{def}}{=} \boldsymbol{X}^{\mathcal{T}} \,|\, \boldsymbol{X}^{\mathcal{T}} \not\leq \boldsymbol{u}$ to be close in distribution to $\boldsymbol{X}_{\boldsymbol{u}}^{\mathcal{G}}$. Hence, we propose to compute their difference in distribution using a simple extension of the EMD in Section 2.3 which is known as the 2-Wasserstein distance [44, Chapter 7]. If $n_{\mathcal{G}}$ and $n_{\mathcal{T}}$ denote, respectively, the size of the test set and the generated set of observations, the measure can be expressed as

$$W_2(\boldsymbol{X}_{\boldsymbol{u}}^{\mathcal{G}}, \boldsymbol{X}_{\boldsymbol{u}}^{\mathcal{T}}) \stackrel{\text{def}}{=} \inf_{\pi \in \Pi\left(\frac{\mathbf{1}_{n_{\mathcal{G}}}}{n_{\mathcal{G}}}, \frac{\mathbf{1}_{n_{\mathcal{T}}}}{n_{\mathcal{T}}}\right)} \sqrt{\sum_{i=1}^{n_{\mathcal{G}}} \sum_{j=1}^{n_{\mathcal{T}}} \|(\boldsymbol{X}_{\boldsymbol{u}}^{\mathcal{G}})_i - (\boldsymbol{X}_{\boldsymbol{u}}^{\mathcal{T}})_j\|_2^2 \, \pi_{ij}} \tag{23}$$

where, for (probability) vectors $\boldsymbol{a} \in \mathbb{R}^k$ and $\boldsymbol{b} \in \mathbb{R}^\ell$, we use the notation

$$\Pi(\boldsymbol{a}, \boldsymbol{b}) \stackrel{\text{def}}{=} \left\{ \pi \in \mathbb{R}_+^{k \times \ell} : \pi \mathbf{1}_\ell = \boldsymbol{a} \text{ and } \pi^T \mathbf{1}_k = \boldsymbol{b} \right\}.$$

We can view π_{ij} as the amount of mass transferred from $(\boldsymbol{X}_{\boldsymbol{u}}^{\mathcal{G}})_i$ to $(\boldsymbol{X}_{\boldsymbol{u}}^{\mathcal{T}})_j$. In a similar way as in definition (13) of the EMD, $W_2(\boldsymbol{X}_{\boldsymbol{u}}^{\mathcal{G}}, \boldsymbol{X}_{\boldsymbol{u}}^{\mathcal{T}})$ measures, among all the possible ways to transport the empirical distribution of the vectors in $\boldsymbol{X}_{\boldsymbol{u}}^{\mathcal{G}}$ to the one of the vectors in $\boldsymbol{X}_{\boldsymbol{u}}^{\mathcal{T}}$, the cheapest way to do so if the cost of unitary transportation is equal to the squared Euclidean distance between points. In the case where we would have normally distributed quantities, this quantity would reduce to computing the well-known *Fréchet*

inception distance, which is popularly used to assess the quality of generated images [23] using generative systems like the GAN, also used in [28] to assess the quality of the MGP output of their algorithm. As we compare quantities that are far from being normally distributed, we think it makes more sense to compute the 2-Wasserstein distance instead.

Architectures. For both the generator and the discriminator in Algorithm 1, we consider fully connected multilayer perceptrons with leaky rectified linear unit activation function [30] which applies the map

$$x \in \mathbb{R} \mapsto \begin{cases} x & \text{if } x \geq 0, \\ \alpha x & \text{else,} \end{cases} \tag{24}$$

to each component of the linear transformation in a given layer, where $\alpha \in [0, 1]$ is typically set to 0.01. Taking $\alpha = 0$ leads to the rectified linear unit activation function and $\alpha = 1$ leads to a simple linear layer. We refer to [30] for a comparison with other standard activation functions. Here, we consider $\alpha = 0.01$ for any hidden layer while we take $\alpha = 1$ for the final layer in each network. More recently, α has been considered as a learnable parameter in the optimization process [22] but this option is not considered in this work. Other architectures are possible if the data have some structure. For example, the authors of [8] consider convolutions in their network as they work with spatial data on a grid, and this could be adapted to our framework too, but here we only consider basic multivariate data and keep this extension for future work.

Hyperparameter search. In addition to the learnable parameters of the generator and discriminator networks in Algorithm 1, many external parameters (hyperparameters) can be chosen by the user. We consider the following values for the hyperparameters:

- batch size $m \in \{\lfloor n/k \rfloor : k = 1, 2, 4, 8, 16\}$;
- dimension of the generator's latent space $\ell \in \{\lfloor (d-1)/k \rfloor : k = 0.25, 0.5, 0.75, 1, 2\}$;
- maximum number of neurons in a hidden layer in $\{32, 64, 128, 256, 512\}$;
- number of hidden layers in $\{1, 2, 4, 8\}$;
- learning rate for the ADAM optimizer in $\{0.01, 0.005, 0.001, 0.0005, 0.0001\}$ and coefficients $\beta = (\beta_1, \beta_2)$ used for computing running averages of gradient and its square in $\{(0.0, 0.9), (0.5, 0.9), (0.5, 0.99)\}$;
- the gradient penalty coefficient $\lambda \in \{0.1, 1, 3, 5, 7, 9\}$;
- the marginal penalty coefficient $\rho \in \{0.001, 0.01, 0.1, 1, 3\}$;
- the number of discriminator iterations per generator iterations $n_D \in \{1, 3, 5, 10\}$.

Inside the space of possible values for those hyperparameters, we perform a random search. Typically, we consider around 2000 different models in this space for a given dataset. We train them with $n_e = 5000$ epochs, since training them even longer did not significantly improve the performance, but greatly affects the computation time. Evaluation of the associated models is based on the extremal coefficient metric $E_{\overline{\theta}}(k)$ in (22) with $k = 2$ and $k = 3$. Validation and final performance assessment of the models are always performed on separate datasets.

Comparison with existing methods. As reviewed in Section 2.1, numerous methods already exist in the rapidly growing field that combines generative approaches with extreme value theory, each with its own advantages and limitations. Therefore, an exhaustive comparison with all existing methods is not feasible, even less so as the software code underlying the numerical results is often not available from the papers. Instead, we choose to compare our approach with versions of two recent methods: HTGAN from [18] and GPGAN from [28]. We discuss our specific implementations in Appendix D.

Software. The code used to produce the results of this section was written in `Python`—in particular, the `PyTorch` [33] framework was used to train the generative models—and in `R` [35], which was mainly used for data preprocessing and computation of the performance measures associated with the generated data, notably via the `transport` [41] package for the computation of the Wasserstein distance W_2 in (23).

Hardware. The training of the various algorithms and the computation of performance metrics have been performed on the CPUs of the Lemaitre4 and the NIC5 computer clusters, which are managed by the Consortium of Équipements de Calcul Intensif (CÉCI). Details and more specifications related to these computers can be found on the CECI website `https://www.ceci-hpc.be/`.

4.2 Simulated data: logistic dependence structure

As an illustration of the method's performance in a controlled setting, we start by considering simulated data. To keep things simple, we consider the same experimental setup as in Section 3.2 of [18], with whom we are going to compare our method.

The random vector $\boldsymbol{X}$ is generated using a d-dimensional Gumbel copula with parameter $\theta \in [1, \infty)$ and Pareto distributed marginals with parameter $\alpha > 0$. It is easy to verify that this data-generating process satisfies Assumptions 1 to 3 in Section 2.2. In particular, the extremal coefficient introduced as a performance measure at the beginning of Section 4 is $\theta_J = |J|^{1/\theta}$. Hence, $\theta \to 1$ corresponds to less dependence in the tail $\boldsymbol{X}$, while $\theta \to \infty$ corresponds to perfect tail dependence.

We consider various scenarios in which we assess the performance of WA-GAN and compare it to HTGAN and GPGAN. For HTGAN we follow [18] and make the unrealistic assumption that α is known. We take $d \in \{10, 20, 50\}$ and $\theta \in \{4/3, 2, 4\}$ leading to a Kendall's tau of $\tau \in \{1/4, 1/2, 3/4\}$. For the margins, we consider $\alpha = 2$ in any scenario.

The models are trained on $n_{\text{train}} = 10\,000$ observations and validated on $n_{\text{val}} = 5\,000$ observations. The performance assessment is done on $n_{\text{test}} = 20\,000$ points.

Results. For each method and each scenario, two scores are computed. On the one hand, a dependence score equal to $\{E_{\overline{\theta}}(2) + E_{\overline{\theta}}(3)\}/2$, with $E_{\overline{\theta}}(k)$ as in (22), is used to validate the hyperparameters and to assess the performance of the tail dependence in the generated data. On the other hand, to assess the quality of generated extremes $\boldsymbol{X} \mid \boldsymbol{X} \not\leq \boldsymbol{u}$, the 2-Wasserstein distance as defined in (23) is computed, where $u_j = \widehat{b}_j(n/k_2)$ with $\widehat{b}_j$ as in (20), for some $k_2 \in \{1, \ldots, n\}$ which satisfies $k_2 \leq k_1$ where $k_1 \in \{1, \ldots, n\}$ is used to train the generator in Algorithm 1. Here we take $n = n_{\text{train}}$ and $k_1 = k_2 = [\sqrt{n}\,]$.

The results are reported in Table 1 below.

		Dependence score			Extremes score		
τ	Model	$d = 10$	$d = 20$	$d = 50$	$d = 10$	$d = 20$	$d = 50$
$\tau = \frac{1}{4}$	WA-GAN	**0.0103**	**0.0074**	**0.0054**	67.311	**62.905**	**42.825**
	HTGAN	0.0146	0.0185	0.0199	1479.9	3106.7	1170.7
	GPGAN	0.0262	0.0321	0.0326	**66.663**	65.429	52.133
$\tau = \frac{1}{2}$	WA-GAN	**0.0176**	0.0207	**0.0097**	**66.734**	**52.832**	**32.468**
	HTGAN	0.0247	**0.0179**	0.0126	2725.3	1447.9	2367.4
	GPGAN	0.0511	0.0671	0.0672	67.841	55.308	46.996
$\tau = \frac{3}{4}$	WA-GAN	**0.0208**	**0.0266**	**0.0158**	**59.548**	**44.602**	**31.808**
	HTGAN	0.0284	0.0338	0.055	891.62	612.44	440.45
	GPGAN	0.0497	0.0726	0.0818	59.919	53.758	45.541

Table 1: Dependence and extremes scores for different dependence τ and dimensions d. For each scenario and each score, the best score (i.e., the lowest) is indicated in bold.

These results show that WA-GAN is competitive with other methods, from the tail dependence perspective as well as from the quality of the generated extremes. The advantage of WA-GAN seems to increase as the dimension d increases. Large values of d is often at the center of interest when generative models for the dependence are used. Note also that it may be surprising at first glance that the scores become better when d increases, but this is probably due to the dependence structure of the data generating process being the same for all pairs of variables, making it easier to learn it when the dimension of the space increases.

Some additional comments can be made for each method. For WA-GAN, the penalization coefficient ρ in (19) is typically selected to be large among the possible values ($\rho = 1$ or $\rho = 3$) showing that it is a good idea to enforce marginal constraints (3) in the generative model. For HTGAN, the obtained values of the hyperparameters match the discussion on pages 13–15 in the paper [18] proposing this method. In particular, we get big batch sizes, big latent space dimensions, and low values of the learning rate included in the range recommendend by the authors. For GPGAN, the training phase is

quicker since, as illustrated on Figure 2, we have a lower training size in this case as this method is trained on points with $|\boldsymbol{V}|_\infty > n/k$ while WA-GAN is trained on points with $|\boldsymbol{V}|_1 > n/k$, which is less restrictive. This can be an asset if we have enough data, but can harm GPGAN's performance if d is large and n is not too big.

As an illustration of the performances of the different methods to accurately reproduce the tail dependence of the data, we plot the empirical bivariate and trivariate extremal coefficients ($|J| = 2$ and $|J| = 3$ in (21)) of the test set against the ones of the training set, and against those of all the generative methods for $d \in \{10, 50\}$ and $\tau = 0.5$. In this setting, we know that the true values of these coefficients are $\sqrt{|J|}$. This is displayed n Figure 3 for $d \in \{10, 50\}$. There is no clear visual difference between WA-GAN and HTGAN, both performing well at representing the coefficients of the test set. GPGAN, however, fails to do so, and even more when the dimension increases. When d gets larger, the empirical extremal coefficients are below their true values, even in the test set. The reason is that the empirical estimator of Φ in (21) becomes less accurate as d increases [9].

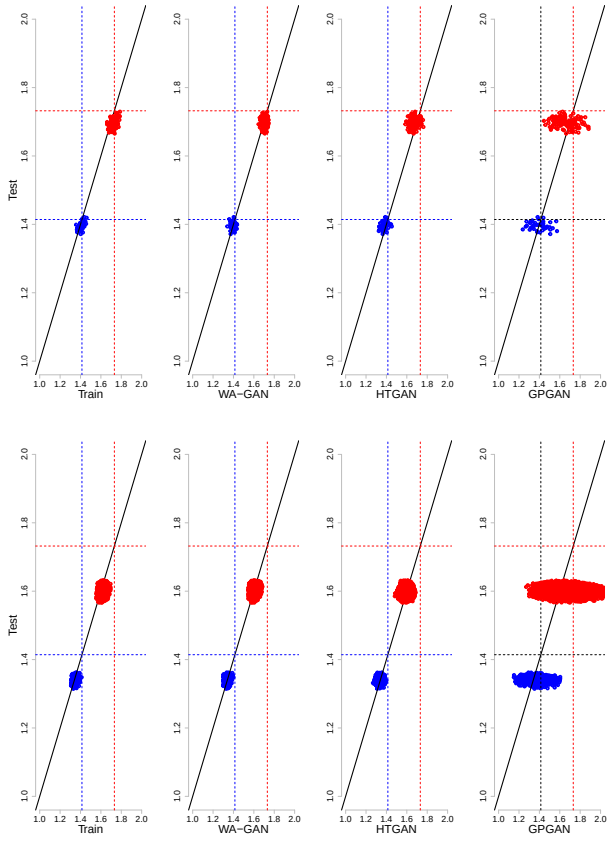


Figure 3: Top: $d = 10$; bottom: $d = 50$. – Extremal coefficients of order $k = 2$ (blue) and $k = 3$ (red). Solid black line is the diagonal. Dashed blue and red lines are the true values of the coefficients for $k = 2$ and $k = 3$ respectively.

Next, to illustrate the accuracy of generated extremes by each method, we plot the projection of each generated sample on the first two margins and compare it to the first two margins of the extremes in the test set. We take again $\tau = 0.5$. Results for are displayed on Figure 4. WA-GAN seems to perform well, whatever the dimension. The results of HTGAN are less satisfying as it is clear from the figure that the angular measure associated with this method is discrete (this corresponds to the "directions" that we observe in the generated extremes) and this does not match the true angular measure which is logistic. This is theoretically justified by Proposition 2.10 in [18] introducing this method, and already visible from their Figure 2 on page 14. GPGAN also performs well even though its performance decreases in comparison to WA-GAN as the dimension increases; see Table 1.

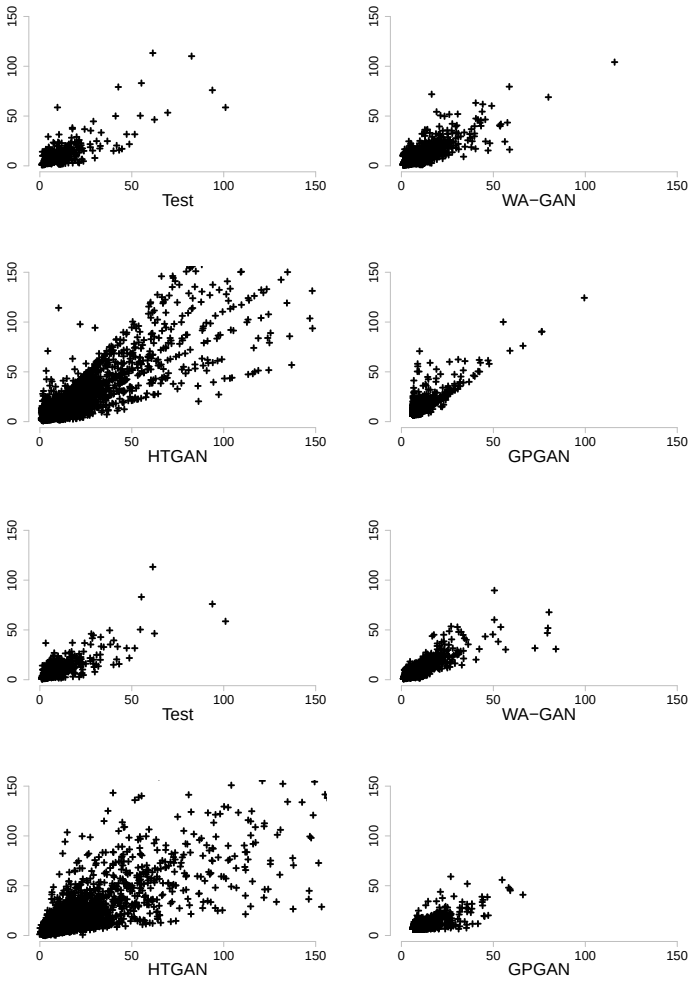


Figure 4: Top: $d = 10$; bottom: $d = 50$. – First two margins of extremes in the test set and the different generative methods.

4.3 Financial data: daily returns of industry portfolios

We apply our methodology to analyze the tail behavior of the "value-averaged" daily returns of $d = 30$ industry portfolios compiled and posted as part of the Kenneth French Data Library. The data in consideration span between 1950 and 2015 with $n = 16\,694$ observations. This data set is very widely used; for analyses in an extreme value context see e.g. [25] and [10]. Details about the portfolio constructions can be found online.[1] Since we are interested in extreme losses we first multiply all returns by -1.

As illustrated by the Kendall's correlation matrix in Figure 5, the daily losses are positively related between the portfolios. We aim to investigate this dependence in the tail part of the distribution.

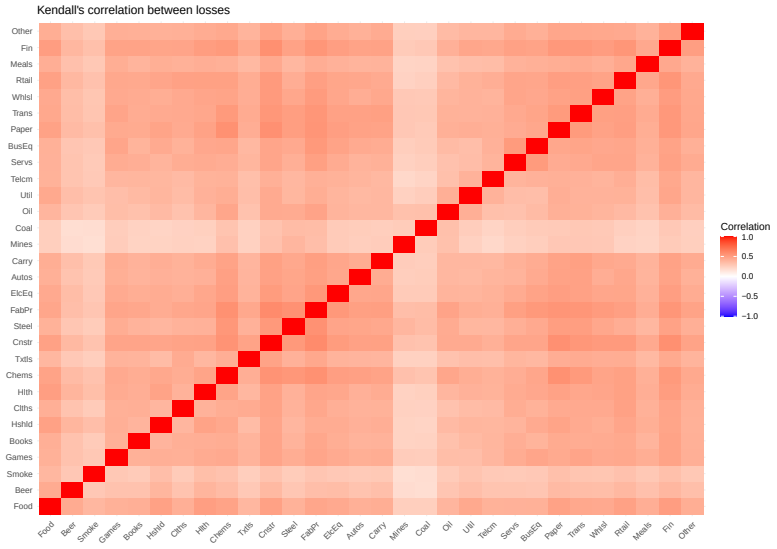


Figure 5: Kendall's correlation matrix between daily losses of the portfolios.

To sample extremes from the joint distribution, we first compute the maximum likelihood estimators of the univariate GPD parameters for each margin. Boxplots illustrating the values of those parameters are reported in Figure 6. The estimated shape values clearly indicates the presence of heavy tails in the marginal distributions. The GPD qq-plots of the marginal fits are to be found on Figures 9, 10 and 11 in Appendix E. They indicate a relatively good fit, even though some extreme quantiles are larger than the ones associated with the GPD model.

For comparison, we start by computing the test scores for WA-GAN, HTGAN and GPGAN. The training and validation procedures are exactly the same as described in Section 4.2. Here, we train on $n_{\text{train}} = 7\,000$, we validate on $n_{\text{val}} = 3\,000$ and compute the scores on $n_{\text{test}} = 6\,694$ observations. We again consider $k = \sqrt{n_{\text{train}}}$ for WA-GAN. For HTGAN, we do not compute the scores on extremes because the marginal distributions

[1] `https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/Data_Library/det_30_ind_port.html`

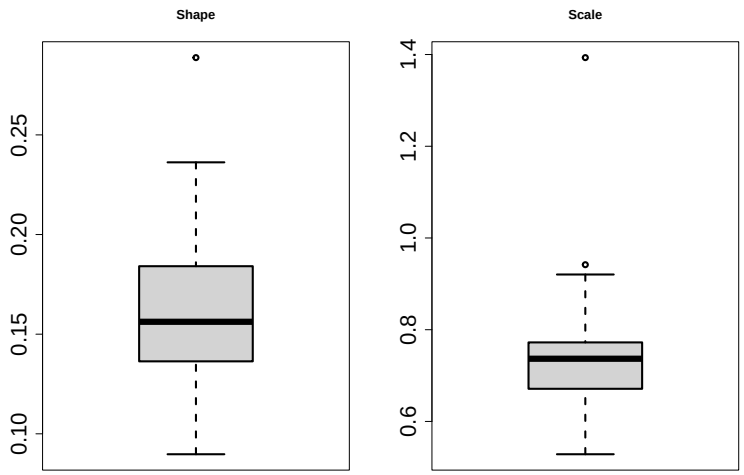


Figure 6: Boxplots of estimated marginal GPD parameters

are not know, see Appendix D. Table 2 shows that WA-GAN seems to be the best at capturing the dependence between extremes in the data while GPGAN has a better 2-Wasserstein distance score.

method	dependence	extremes
WA-GAN	**0.018**	11.93
HTGAN	0.026	/
GPGAN	0.043	**8.46**

Table 2: Test scores of each method on the $d = 30$ financial dataset. The best score (i.e., the lowest) is indicated in bold.

As already pointed out in the analysis of [25], the extreme losses of those portfolios can be clustered into different kind of industries. For example, the tobacco industry exhibits asymptotic independence to all other categories (e.g., to the textile industries) while the extreme losses of the business and IT-related industries are dependent. We illustrate the fact that WA-GAN also captures those phenomena in Figure 7 by displaying the two-dimensional projections of the estimated angular measure on those margins. We see that the generated "angles" associated with tobacco/textile portfolios are much more concentrated around the axes than for business/IT, showing that WA-GAN captures this complex tail dependence.

As a further illustration we consider a portfolio obtained by combining the mines industry portfolio with the coal industry portfolio (with equal weight). It is intuitive and has been shown in [10, 25] that the mines and coal industries exhibit asymptotic dependence. If one wants to sample the losses associated to the portfolio we propose,

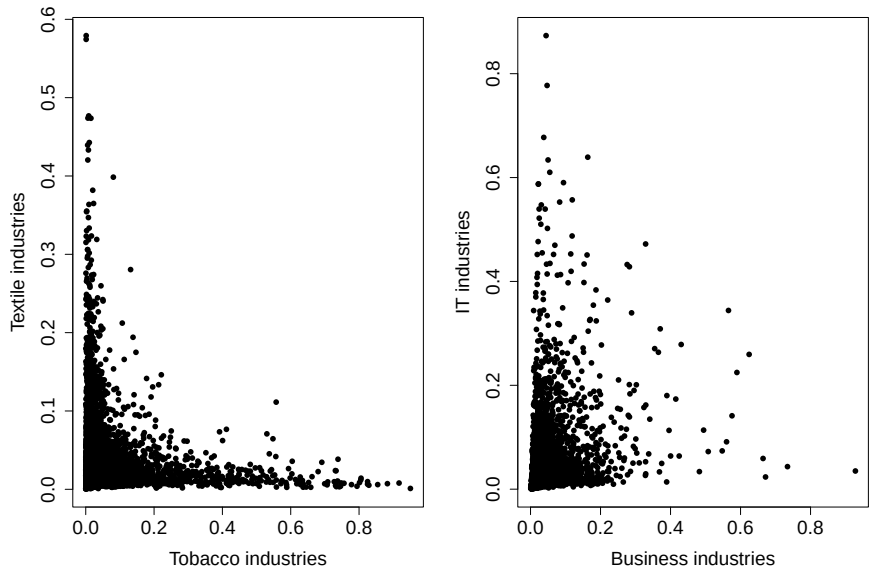


Figure 7: Two-dimensional projections of the generated "angles" from Φ by WA-GAN.

this tail dependence should be taken into account. Figure 8 represents the densities of simulated values of the extreme losses of the combined portfolio (i.e., both marginal losses exceeds a large threshold) based on WA-GAN (in black) and based on independent simulations from the marginal GPDs (in blue). Not taking the tail dependence into account leads to underestimation of the risk associated with this portfolio.

5 Conclusions

This paper develops the WA-GAN method which is built on the L_1-norm and on the asymptotic independence of the angular and radial parts of a d-dimensional regularly varying random vector. An important component is the use of Aitchison coordinates which transforms values on the unit simplex in $\mathbb{R}^d$ to the entire linear space $\mathbb{R}^{d-1}$. The method provides simulated extreme values and can hence be used to estimate probabilities of extreme events, even more extreme than those already observed.

The method is tried out on simulated extreme value data sets of dimensions $d = 10$, 20 and 50 with 10 000 observations in the training set and 5 000 observations in the validation set. Plots and two performance measures which (i) score the dependence between extremes using extremal coefficients and (ii) score the quality of the simulated values on the real scale using the 2-Wasserstein distance show that WA-GAN performs well. Further, in Table 1 WA-GAN is shown to have better values of these performance scores than the HTGAN and GPGAN methods.

The methods are also applied to a financial data set from the Kenneth French Data

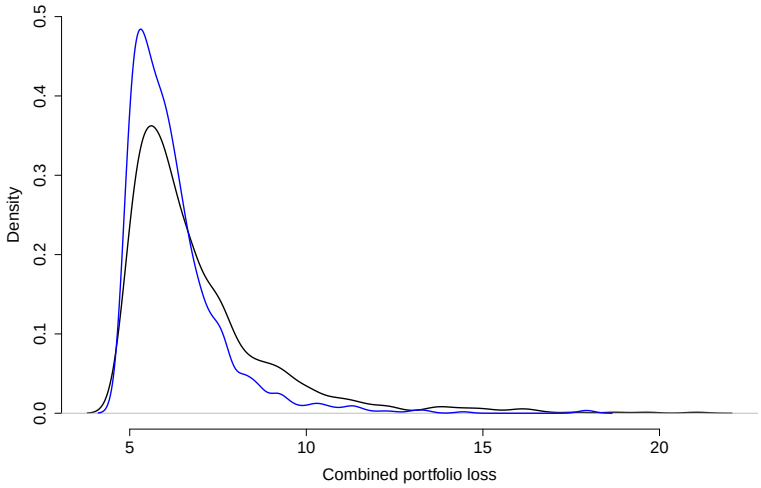


Figure 8: Densities of extreme losses in the combined portfolio of mines and coals industries based on WA-GAN (black) and independently simulated margins (blue).

Library. Again WA-GAN performs well and underlines the importance of being able to handle dependence between different financial portfolios. In terms of performance measures, WA-GAN has the best dependence score, while GPGAN has a better 2-Wasserstein score than WA-GAN.

In this paper we do not quantify the uncertainty in the estimated probability of an extreme event. However, provided enough computational power is available, this could be done using a approach similar to a parametric bootstrap. For this one would repeat the following, say, 100 times: first use the fitted WA-GAN to simulate a new extreme data set of the same size as the original one, then fit a new WA-GAN to this simulated data set and use the new WA-GAN to estimate the probability of interest. This will provide 100 estimated probability values which can be used to find "confidence bounds" for the original estimated probability.

Code and data availability

The code is available from the first author and will be made publicly available in a software repository. The data underlying the analysis in Section 4.3 is publicly available from the Kenneth R. French data library `https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html`.

Acknowledgments

The research of Stéphane Lhaut was supported by the Fonds National de la Recherche Scientifique (FNRS, Belgium) within the framework of a FRIA grant (No. 1.E.114.23F). Computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region.

References

[1] John Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society serie B*, 44(2):139–177, 1982.

[2] Michaël Allouche, Stéphane Girard, and Emmanuel Gobet. Ev-gan: Simulation of extreme events with relu neural networks. *Journal of Machine Learning Research*, 23(150):1–39, 2022.

[3] Michaël Allouche, Stéphane Girard, and Emmanuel Gobet. On the simulation of extreme events with neural networks. *HAL preprint, hal-04416809v2*, pages 1–35, 2024.

[4] Hugues Annoye. *Some contributions to synthetic data creation.* PhD thesis, UCLouvain, 2024.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017.

[6] August A. Balkema and Laurens de Haan. Residual life time at great age. *The Annals of Probability*, 2(5):792–804, 1974.

[7] Jan Beirlant, Yuri Goegebeur, Johan Segers, and Jozef L. Teugels. *Statistics of Extremes: Theory and Applications.* Wiley Series in Probability and Statistics. Wiley, 2004.

[8] Younes Boulaguiem, Jakob Zscheischler, Edoardo Vignotto, Karin van der Wiel, and Sebastian Engelke. Modeling and simulating spatial extremes by combining extreme value theory with generative adversarial networks. *Environmental Data Science*, 1:1–18, 2022.

[9] Stéphan Clémençon, Hamid Jalalzai, Stéphane Lhaut, Anne Sabourin, and Johan Segers. Concentration bounds for the empirical angular measure with statistical learning applications. *Bernoulli*, 29(4):2797–2827, 2023.

[10] Daniel Cooley and Emeric Thibaud. Decompositions of dependence for high-dimensional extremes. *Biometrika*, 106(3):587–604, 2019.

[11] Laurens de Haan and Ana Ferreira. *Extreme Value Theory: An Introduction*. Springer Series in Operations Research and Financial Engineering. Springer New York, NY, 2006.

[12] Laurens de Haan and Sidney I. Resnick. Limit theory for multivariate sample extremes. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 40:317–337, 1977.

[13] Juan J. Egozcue, Vera Pawlowsky-Glahn, Gloria Mateu-Figueras, and Carles Barceló-Vidal. Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, 35(3):279–300, 2003.

[14] John H.J. Einmahl, Laurens de Haan, and Vladimir I. Piterbarg. Nonparametric estimation of the spectral measure of an extreme value distribution. *Annals of Statistics*, 29(5):1401–1423, 2001.

[15] John H.J. Einmahl, Laurens de Haan, and Ashoke Kumar Sinha. Estimating the spectral measure of an extreme value distribution. *Stochastic Processes and their Applications*, 70(2):143–171, 1997.

[16] John H.J. Einmahl and Johan Segers. Maximum empirical likelihood estimation of the spectral measure of an extreme-value distribution. *Annals of Statistics*, 37(5B):2953–2989, 2009.

[17] Ana Ferreira and Laurens de Haan. The generalized Pareto process; with a view towards application and simulation. *Bernoulli*, 20(4):1717 – 1737, 2014.

[18] Stéphane Girard, Emmanuel Gobet, and Jean Pachebat. Deep generative modeling of multivariate dependent extremes. *HAL preprint, hal-04700084v2*, pages 1–35, 2024.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua 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. Curran Associates, Inc., 2014.

[20] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[21] Ali Hasan, Khalil Elkhalil, Yuting Ng, João M. Pereira, Sina Farsiu, Jose H. Blanchet, and Vahid Tarokh. Modeling extremes with d-max-decreasing neural networks. In *Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 180 of *Proceedings of Machine Learning Research*, pages 759–768. PMLR, 2022.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[23] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[24] Callum J. R. Murphy-Barltrop, R. Majumder, and J. Richards. Deep Learning of Multivariate Extremes via a Geometric Representation. *arXiv:2406.19936v2*, pages 1–66, 2024.

[25] Anja Janßen and Phyllis Wan. k-means clustering of extremes. *Electronic Journal of Statistics*, 14(1):1211–1233, 2020.

[26] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[27] Nicolas Lafon, Philippe Naveau, and Ronan Fablet. A vae approach to sample multivariate extremes. *arXiv preprint arXiv:2306.10987*, 2023.

[28] Jiawei Li, Dong Li, Peng Li, and Gennady Samorodnitsky. Generalized pareto gan: Generating extremes of distributions. In *Proceedings of the 2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024.

[29] Erik Linder-Norén. PyTorch–GAN, 2025. Available on: `https://github.com/eriklindernoren/PyTorch-GAN`.

[30] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013. Workshop on Deep Learning for Audio, Speech and Language Processing.

[31] Anas Mourahib, Anna Kiriliouk, and Johan Segers. Multivariate generalized Pareto distributions along extreme directions. *Extremes*, pages 10.1007/s10687–024–00501–4, 2024.

[32] Philippe Naveau and Johan Segers. Multivariate extreme value theory. *arXiv preprint*, pages 1–33, 2024.

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In

H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[34] James Pickands III. Statistical inference using extreme order statistics. *The Annals of Statistics*, 3(1):119–131, 1975.

[35] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2023.

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

[37] Sidney I. Resnick. *Heavy-Tail Phenomena: Probabilistic and Statistical Modeling*. Springer Series in Operations Research and Financial Engineering. Springer, 2007.

[38] Holger Rootzén, Johan Segers, and Jennifer L. Wadsworth. Multivariate generalized Pareto distributions: Parametrizations, representations, and properties. *Journal of Multivariate Analysis*, 165:117–131, 2018.

[39] Holger Rootzén, Johan Segers, and Jennifer L. Wadsworth. Multivariate peaks over thresholds models. *Extremes*, 21(1):115–145, 2018.

[40] Holger Rootzén and Nader Tajvidi. Multivariate generalized pareto distributions. *Bernoulli*, 12(5):917–930, 2006.

[41] Dominic Schuhmacher, Björn Bähre, Nicolas Bonneel, Carsten Gottschlich, Valentin Hartmann, Florian Heinemann, Bernhard Schmitzer, and Jörn Schrieber. *transport: Computation of Optimal Transport Plans and Wasserstein Distances*, 2024. R package version 0.15-4.

[42] Richard L Smith. Estimating tails of probability distributions. *The Annals of Statistics*, 15(3):1174–1207, 1987.

[43] Aad W. van der Vaart and Jon A. Wellner. *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer Series in Statistics. Springer, New York, NY, 1996.

[44] Cédric Villani. *Topics in Optimal Transportation*, volume 58 of *Graduate Studies in Mathematics*. American Mathematical Society, 2003.

A Proof of Proposition 1

Proof. We start by noting that $\boldsymbol{X} \overset{\mathcal{L}}{=} \boldsymbol{b}(\boldsymbol{V})$ and, by the continuity of the margins F_j, each tail quantile function b_j is strictly increasing, so that $\{\boldsymbol{X} \not\leq \boldsymbol{b}(t)\} = \{\boldsymbol{V} \not\leq t\}$ for any $t > 1$. Consequently,

$$\frac{\boldsymbol{X} - \boldsymbol{b}(t)}{\boldsymbol{a}(t)} \,\Big|\, \boldsymbol{X} \not\leq \boldsymbol{b}(t) \overset{\mathcal{L}}{=} \frac{\boldsymbol{b}(t(t^{-1}\boldsymbol{V})) - \boldsymbol{b}(t)}{\boldsymbol{a}(t)} \,\Big|\, \boldsymbol{V} \not\leq t.$$

Furthermore, it follows easily from Assumption 1 that

$$\left(t^{-1}\boldsymbol{V} \mid \boldsymbol{V} \not\leq t\right) \overset{\mathcal{L}}{\to} \boldsymbol{Y}, \qquad t \to \infty, \tag{25}$$

where, for Borel set $B \subseteq \mathbb{E}$, writing $\mathbb{L}_1 \overset{\text{def}}{=} \{\boldsymbol{x} \in [0,\infty)^d : \boldsymbol{x} \not\leq 1\}$, we have

$$\mathsf{P}(\boldsymbol{Y} \in B) = \frac{\nu(B \cap \mathbb{L}_1)}{\nu(\mathbb{L}_1)}. \tag{26}$$

The exponent measure ν is connected to the angular measure Φ introduced in (2). More precisely [7, Chapter 8], for any bounded, measurable function $f : \mathbb{E} \to \mathbb{R}$ that is zero in a neighborhood of the origin, we have

$$\nu(f) \overset{\text{def}}{=} \int_{\mathbb{E}} f(x)\,\mathrm{d}\nu(x) = d\int_{\Delta_{d-1}} \int_0^\infty f(y\boldsymbol{w}) \frac{\mathrm{d}y}{y^2}\,\mathrm{d}\Phi(\boldsymbol{w}). \tag{27}$$

In particular, for any for Borel set $B \subseteq \mathbb{E}$,

$$\begin{aligned} d^{-1}\nu(B \cap \mathbb{L}_1) &= \int_{\Delta_{d-1}} \int_0^\infty \mathbb{I}_{B\cap\mathbb{L}_1}(y\boldsymbol{w}) \frac{\mathrm{d}y}{y^2}\,\mathrm{d}\Phi(\boldsymbol{w}) \\ &= \int_{\Delta_{d-1}} \int_1^\infty \mathbb{I}_{B\cap\mathbb{L}_1}(y\boldsymbol{w}) \frac{\mathrm{d}y}{y^2}\,\mathrm{d}\Phi(\boldsymbol{w}) \\ &= \mathsf{P}(Y\boldsymbol{\Theta} \in B \cap \mathbb{L}_1), \end{aligned}$$

where Y is unit-Pareto distributed and independent of $\boldsymbol{\Theta} \sim \Phi$. Whence, by (26),

$$\mathsf{P}(\boldsymbol{Y} \in B) = \frac{d\,\mathsf{P}(Y\boldsymbol{\Theta} \in B \cap \mathbb{L}_1)}{d\,\mathsf{P}(Y\boldsymbol{\Theta} \in \mathbb{L}_1)} = \mathsf{P}\left(Y\boldsymbol{\Theta} \in B \mid Y\boldsymbol{\Theta} \in \mathbb{L}_1\right),$$

From the local uniformity of the convergence in point (ii) of Remark 1, guaranteed by Assumption 3, and the fact that $\mathsf{P}(Y_j > 0) = 1$ for any $j \in \{1,\ldots,d\}$ by the previous computations and Assumption 2, the extended continuous mapping theorem [43, Theorem 1.11.1] applies, yielding

$$\frac{\boldsymbol{b}(t(t^{-1}\boldsymbol{V})) - \boldsymbol{b}(t)}{\boldsymbol{a}(t)} \,\Big|\, \boldsymbol{V} \not\leq t \overset{\mathcal{L}}{\to} \frac{\boldsymbol{Y}^{\boldsymbol{\xi}} - 1}{\boldsymbol{\xi}}, \qquad t \to \infty.$$

This concludes the proof of (7) and (8). Equation (9) is a direct consequence of (7) by noting that

$$\sigma_j(u_j(t)) = a_j\left(\frac{1}{1 - F_j(b_j(t))}\right) = a_j(t), \qquad t > 1$$

since $F_j(F_j^{-1}(p)) = p$ for all $0 < p < 1$ by continuity of F_j. □

B The Aitchison simplex

The open simplex Δ°_{d-1} is equipped with an inner product space structure with the following operations [1]: for $\boldsymbol{v} = (v_1, \dots, v_d), \boldsymbol{w} = (w_1, \dots, w_d) \in \Delta^{\circ}_{d-1}$ and $\alpha \in \mathbb{R}$, define the following operations:

- Addition:
$$\boldsymbol{v} \oplus \boldsymbol{w} \stackrel{\text{def}}{=} \left(\frac{v_j w_j}{\sum_{i=1}^d v_i w_i} \right)_{j=1}^d$$
- Scalar multiplication:
$$\alpha \odot \boldsymbol{v} \stackrel{\text{def}}{=} \left(\frac{v_j^\alpha}{\sum_{i=1}^d v_i^\alpha} \right)_{j=1}^d$$
- (Aitchison) inner product:
$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle_A \stackrel{\text{def}}{=} \sum_{i=1}^d \log \left(\frac{v_i}{g(\boldsymbol{v})} \right) \log \left(\frac{w_i}{g(\boldsymbol{w})} \right),$$
where $g : \boldsymbol{u} \in \Delta^{\circ}_{d-1} \mapsto g(\boldsymbol{u}) \stackrel{\text{def}}{=} (\prod_{i=1}^d u_i)^{1/d}$ is the geometric mean.

It is an easy exercise to verify that $(\Delta^{\circ}_{d-1}, \oplus, \odot)$ forms a real vector space with zero element $\mathbf{0}_A = \mathbf{1}_d/d \in \Delta^{\circ}_{d-1}$, where $\mathbf{1}_d = (1, \dots, 1) \in \mathbb{R}^d$, and that $\langle \cdot, \cdot \rangle_A$ defines an inner product on it. The open simplex equipped with this structure is often called the *Aitchison simplex*.

A trivial but useful observation is that if one introduces the so-called *Centered LogRatio (CLR)* operator

$$\operatorname{clr} : \boldsymbol{u} \in \Delta^{\circ}_{d-1} \mapsto \operatorname{clr}(\boldsymbol{u}) \stackrel{\text{def}}{=} \left(\log \left(\frac{u_j}{g(\boldsymbol{u})} \right) \right)_{j=1}^d \in \mathbb{H} \subset \mathbb{R}^d, \tag{28}$$

where $\mathbb{H} \stackrel{\text{def}}{=} \{ \boldsymbol{x} \in \mathbb{R}^d : \langle \boldsymbol{x}, \mathbf{1}_d \rangle = 0 \}$ with $\langle \cdot, \cdot \rangle$ being the standard inner product in $\mathbb{R}^d$, then

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle_A = \langle \operatorname{clr}(\boldsymbol{v}), \operatorname{clr}(\boldsymbol{w}) \rangle, \qquad \boldsymbol{v}, \boldsymbol{w} \in \Delta_{d-1},$$

so that clr naturally defines an isometry between $(\Delta^{\circ}_{d-1}, \langle \cdot, \cdot \rangle_A)$ and $(\mathbb{H}, \langle \cdot, \cdot \rangle)$. The inverse transformation $\operatorname{clr}^{-1} : \mathbb{H} \mapsto \Delta^{\circ}_{d-1}$ is the well known *softmax* function

$$\operatorname{clr}^{-1}(\boldsymbol{x}) = \left(\frac{\exp(\boldsymbol{x}_j)}{\sum_{i=1}^d \exp(\boldsymbol{x}_i)} \right)_{j=1}^d = \operatorname{softmax}(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{H}.$$

The above considerations lead to a clear and easy way to construct an orthonormal basis of the Aitchison simplex:

1. Take any linearly independent family $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d-1}\}$ in $\mathbb{H}$.
2. Apply the Gram–Schmidt procedure to produce an orthonormal basis $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{d-1}\}$ of $\mathbb{H}$ with respect to the standard inner product on $\mathbb{R}^d$.
3. Compute $\{\boldsymbol{e}_i^* \stackrel{\text{def}}{=} \mathrm{clr}^{-1}(\boldsymbol{e}_i) : i = 1, \ldots, d-1\}$, which forms an orthornormal basis of the Aitchison simplex.

Applying the above procedure to the free family

$$\left\{\boldsymbol{x}_i \stackrel{\text{def}}{=} (0, \ldots, 0, 1, -1, 0, \ldots, 0) \in \mathbb{H} : i = 1, \ldots, d-1\right\},$$

where in $\boldsymbol{x}_i$ the 1 element lies at the i-th position, leads to the orthonormal basis of the Aitchison simplex presented in Proposition 2.

C Angular measure with respect to another norm

The angular measure Φ introduced in (2) can be introduced with respect to an arbitrary norm $\|\cdot\|$ on $\mathbb{R}^d$, that is, if we let

$$\widetilde{R} = \|\boldsymbol{V}\| \qquad \text{and} \qquad \widetilde{\boldsymbol{W}} = \widetilde{R}^{-1}\boldsymbol{V},$$

one can consider the measure $\widetilde{\Phi}$ on $\widetilde{\Delta}_{d-1} \stackrel{\text{def}}{=} \{\boldsymbol{x} \in [0,1]^d : \|\boldsymbol{x}\| = 1\}$ obtained by taking, as in (2),

$$\widetilde{\Phi}(A) = \lim_{t\to\infty} \mathsf{P}\left(\widetilde{\boldsymbol{W}} \in A \mid \widetilde{R} \geq t\right), \tag{29}$$

for any Borel set $A \subseteq \widetilde{\Delta}_{d-1}$ such that $\widetilde{\Phi}(\partial A) = 0$. Popular choices for $\|\cdot\|$ consist of the L_p norms $|\cdot|_p$ on $\mathbb{R}^d$ for $p \in [1, \infty]$, see, e.g., [14, 16, 9]. Even though we only consider the norm $|\cdot|_1$ in Algorithm 1, we show that a simple post-processing step permits to provide an estimate for $\widetilde{\Phi}$ also.

From (29), one may show that $\widetilde{\Phi}(A) = \Psi(A)/\Psi(\widetilde{\Delta}_{d-1})$ where Ψ is the finite Borel measure on $\widetilde{\Delta}_{d-1}$ obtained by taking

$$\Psi(A) = \lim_{t\to\infty} t\ \mathsf{P}\left(\widetilde{\boldsymbol{W}} \in A, \widetilde{R} \geq t\right)$$

provided $\Psi(\partial A) = 0$. From (27), for any bounded, measurable $f : \mathbb{E} \to \mathbb{R}$ vanishing in a neighborhood of $\mathbf{0}$, we get

$$d\int_{\Delta_{d-1}} \int_0^\infty f(r\boldsymbol{w}) \frac{\mathrm{d}r}{r^2}\, \mathrm{d}\Phi(\boldsymbol{w}) = \int_{\widetilde{\Delta}_{d-1}} \int_0^\infty f(r\widetilde{\boldsymbol{w}}) \frac{\mathrm{d}r}{r^2}\, \mathrm{d}\Psi(\widetilde{\boldsymbol{w}}),$$

Taking $f(x) = g(x/\|x\|)\mathbb{I}(\|x\| \geq 1)$ for $x \in \mathbb{E}$, where $g : \widetilde{\Delta}_{d-1} \to \mathbb{R}$ is some bounded function, we get

$$d\int_{\Delta_{d-1}} g\left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right) \|\boldsymbol{w}\| \mathrm{d}\Phi(\boldsymbol{w}) = \int_{\widetilde{\Delta}_{d-1}} g(\widetilde{\boldsymbol{w}}) \mathrm{d}\Psi(\widetilde{\boldsymbol{w}}).$$

Picking $g(\widetilde{\boldsymbol{w}}) = \mathbb{I}(\widetilde{\boldsymbol{w}} \in \widetilde{\Delta}_{d-1})$ gives

$$\Psi(\widetilde{\Delta}_{d-1}) = d \int_{\Delta_{d-1}} \|\boldsymbol{w}\| \, \mathrm{d}\Phi(\boldsymbol{w}),$$

so that for any bounded $g : \widetilde{\Delta}_{d-1} \to \mathbb{R}$ we have

$$\int_{\widetilde{\Delta}_{d-1}} g(\widetilde{\boldsymbol{w}}) \, \mathrm{d}\widetilde{\Phi}(\widetilde{\boldsymbol{w}}) = \frac{\int_{\Delta_{d-1}} g\left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right) \|\boldsymbol{w}\| \, \mathrm{d}\Phi(\boldsymbol{w})}{\int_{\Delta_{d-1}} \|\boldsymbol{w}\| \, \mathrm{d}\Phi(\boldsymbol{w})}. \tag{30}$$

In particular, $\widetilde{\Phi}$ is fully determined by Φ.

Suppose that we observe $\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_n \sim \Phi$ as it would be (approximately) the case via the trained generator in Algorithm 1. They lead to the empirical estimator $\widehat{\Phi} = (1/n) \sum_{i=1}^n \delta_{\boldsymbol{\Theta}_i}$ for Φ. Relation (30) then justifies the approximation

$$\begin{aligned}\int_{\widetilde{\Delta}_{d-1}} g(\widetilde{\boldsymbol{w}}) \, \mathrm{d}\widetilde{\Phi}(\widetilde{\boldsymbol{w}}) &\approx \frac{\int_{\Delta_{d-1}} g\left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right) \|\boldsymbol{w}\| \, \mathrm{d}\widehat{\Phi}(\boldsymbol{w})}{\int_{\Delta_{d-1}} \|\boldsymbol{w}\| \, \mathrm{d}\widehat{\Phi}(\boldsymbol{w})} \\ &= \sum_{i=1}^n \Lambda_i g\left(\frac{\boldsymbol{\Theta}_i}{\|\boldsymbol{\Theta}_i\|}\right) = \int_{\widetilde{\Delta}_{d-1}} g(\widetilde{\boldsymbol{w}}) \, \mathrm{d}\widehat{\widetilde{\Phi}}(\widetilde{\boldsymbol{w}}),\end{aligned}$$

where

$$\widehat{\widetilde{\Phi}} \stackrel{\text{def}}{=} \sum_{i=1}^n \Lambda_i \delta_{\boldsymbol{\Theta}_i/\|\boldsymbol{\Theta}_i\|}, \qquad \Lambda_i \stackrel{\text{def}}{=} \frac{\|\boldsymbol{\Theta}_i\|}{\sum_{i=1}^n \|\boldsymbol{\Theta}_i\|}.$$

Said otherwise, a sample $\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_n \sim \Phi$ can be used to produce an estimator for $\widetilde{\Phi}$ by considering the weighted empirical distribution of the rescaled angles $\boldsymbol{\Theta}_i$ on $\widetilde{\Delta}_{d-1}$, where the weights are proportional to the norm $\|\boldsymbol{\Theta}\|_i$.

D Implementation of competing methods

HTGAN is follows a standard GAN architecture [19] with heavy-tailed latent variables that are Pareto(α) distributed for some $\alpha > 0$. It does not make use of (multivariate) extreme value theory at all and rests on the capacity of the GAN to accurately capture the dependence of the data, also in the tail. As advised in their paper, we base our implementation on the one in [29]. Their preprocessing step in their Algorithm 1 on page 10, equation (3.11), requires an estimate of $\gamma = 1/\alpha$ which is not provided in their methodology. They use the true value in their simulations, which is why we will also do so here, in Section 4.2. For real data applications, we will consider $\gamma = 1/1.5$ as recommended by the authors in their results on page 19. Note also that it is not clear to us how step (3.12) in their Algorithm 2 is supposed to work as $\tilde{X}_j^{(i)}$ is not an element of $[0, 1]$ so that it does not make sense to apply the inverse of the estimated marginal distribution function F_j. Since in this simulation setting the output of their generator is

supposed to live in the same space as the original data, we do not apply any transform to compute the score (23). The score (22) is, of course, not affected by any marginal transformation. In real data applications, we do not compute the W_2 score and only consider the dependence score as done in the original paper. In terms of architectures and hyperparameters, we consider the same setting as for our method. This is very close to what the authors also do in their paper, besides the hyperparameter search which is performed through a Bayesian search in their numerical section. Both the generator and the discriminator are trained on 5000 epochs. The final layer of the discriminator has a sigmoid activation function, as it is commonly applied in GAN architectures.

The GPGAN is closer to our approach as it is also based on the MGP distribution to accurately sample from the tail of $\boldsymbol{X}$, even though the approach in [28] is quite different. The authors also rely on a GAN architecture so that we also rely on [29] to implement their methodology. The main differences between their approach and ours are as follows:

- In [28], the dependence is captured by the spectral vector $\boldsymbol{S}$ as introduced in (11), while our techniques rely on sampling from Φ after a transformation to the Aitchison coordinates which helps simplifying the distributions.
- The discriminator in the GPGAN procedure directly evaluates sample quality on the MGP scale (after evaluating the marginal parameters), while our technique separates the dependence structure, for which we rely on the WGAN architecture (Algorithm 1), and the marginal considerations that permit to transform the output back to the MGP scale (Algorithm 2). Our approach has the advantage that if one is only interested in evaluating a tail dependence measure, like the extremal coefficients in (22), it is possible to do so without putting any assumptions on the marginal tail behavior and to work only with Algorithm 1.

The adaptive procedure in [28] to pick the threshold $\boldsymbol{u}$ based on the "extremeness" function E was not entirely clear to us. Therefore, we decided to pick the same $\boldsymbol{u}$ as we do to compute the marginal parameters, so that we work with the same estimates of the parameters $\boldsymbol{\sigma}$ and $\boldsymbol{\xi}$. Also, as in [28] the latent distribution is not indicated, we consider a multivariate standard normal distribution on the latent space. The architectures and hyperparameters are also treated in the same way as for our method and the HTGAN method. Both the generator and the discriminator are trained on 5000 epochs, based on the Adam optimizer, as done in the original paper [28]. The final layer of the discriminator has a sigmoid activation function, as it is commonly applied in GAN architectures. Note that computing the score (23) poses no problem with this method as it outputs directly on the scale of $\boldsymbol{X} \mid \boldsymbol{X} \not\leq \boldsymbol{u}(t)$; however, for the score (22), one needs to be more cautious. For this, we rely on the fact that if $\boldsymbol{X}_1^*, \dots, \boldsymbol{X}_n^*$ forms a sample from $\boldsymbol{X} \mid \boldsymbol{X} \not\leq \boldsymbol{u}(t)$, then

$$\boldsymbol{V}_i^* \overset{\text{def}}{=} \boldsymbol{b}(\boldsymbol{X}_i^*) \sim \boldsymbol{V} \mid \boldsymbol{V} \not\leq t = \boldsymbol{V} \mid |\boldsymbol{V}|_\infty > t, \qquad i \in \{1, \dots, n\}.$$

For $t > 1$ large enough, it means that

$$\boldsymbol{\Theta}_i^* \overset{\text{def}}{=} \frac{\boldsymbol{V}_i^*}{|\boldsymbol{V}_i^*|_\infty} \overset{\mathcal{L}}{\approx} \Phi_\infty, \qquad i \in \{1, \dots, n\}$$

where Φ_∞ the angular measure with respect to the L_∞-norm. Relying on the change of norm explained in Appendix C, we get an estimate for Φ by considering the discrete measure

$$\widehat{\Phi}^* \stackrel{\text{def}}{=} \sum_{i=1}^{n} \Lambda_i^* \delta_{\frac{\boldsymbol{\Theta}_i^*}{|\boldsymbol{\Theta}_i^*|_1}}, \qquad \Lambda_i^* \stackrel{\text{def}}{=} \frac{|\boldsymbol{\Theta}_i^*|_1}{\sum_{i=1}^{n} |\boldsymbol{\Theta}_i^*|_1}.$$

The estimator can be plugged in the formula of extremal coefficient (21) which is used in computing the score (22).

E Additional figures

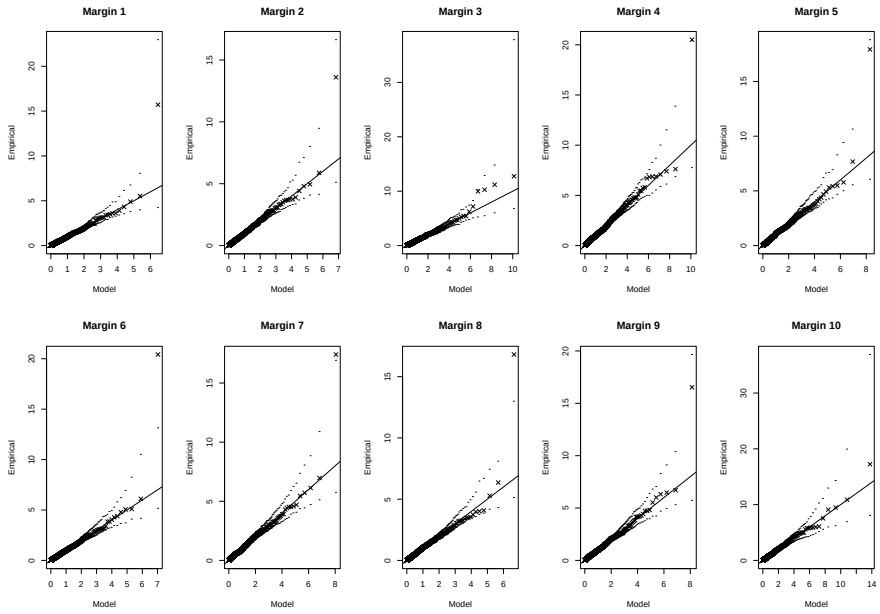


Figure 9: GPD-qqplots for margins 1 to 10 in the financial dataset.

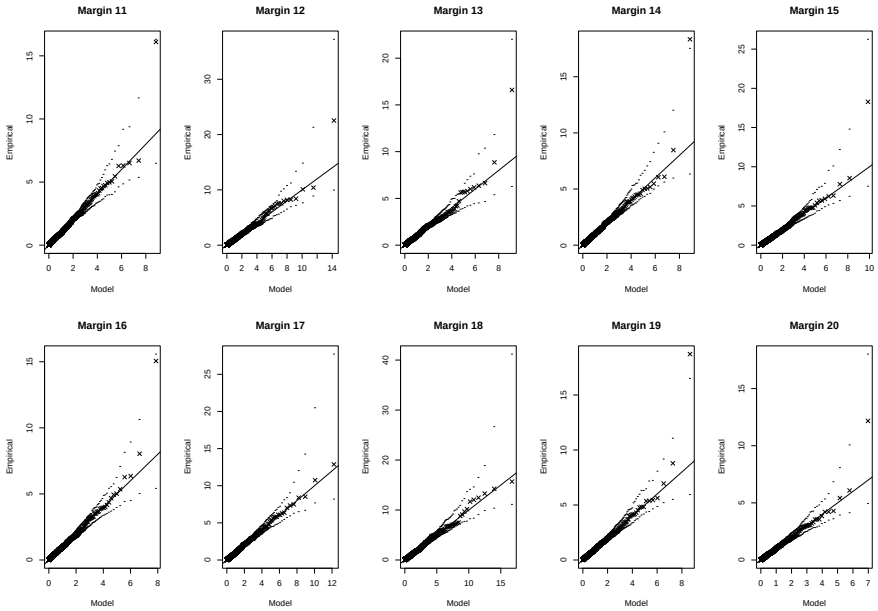


Figure 10: GPD-qqplots for margins 11 to 20 in the financial dataset.

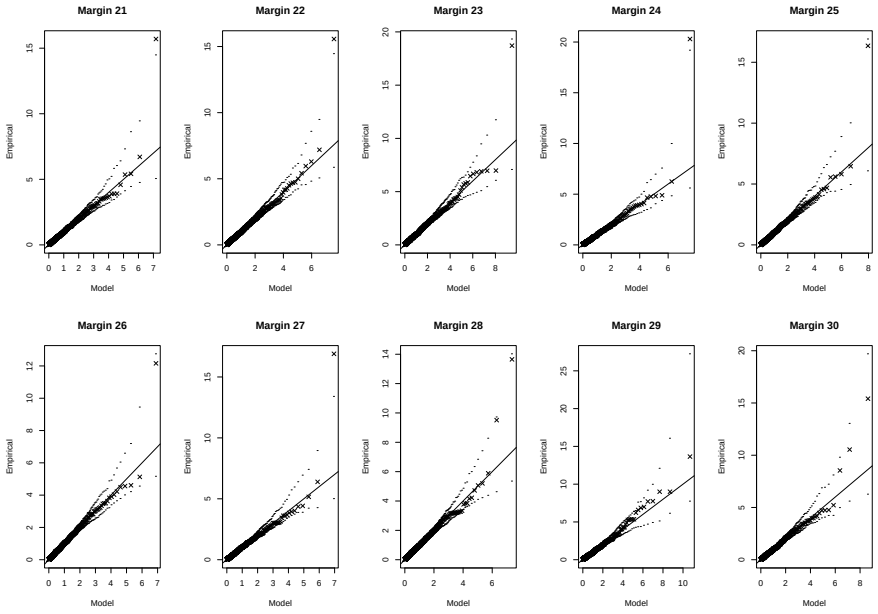


Figure 11: GPD-qqplots for margins 21 to 30 in the financial dataset.