On the simulation of extreme events with neural networks

Michaël Allouche, Stéphane Girard, Emmanuel Gobet

On the simulation of extreme events with neural networks

Michaël Allouche[(1)], Stéphane Girard[(2,⋆)] and Emmanuel Gobet[(3)]

[(1)] Kaiko - Quantitative Data. 2 rue de Choiseul 75002 Paris, France.
[(2)] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France.
[(⋆)] Corresponding author: Stephane.Girard@inria.fr
[(3)] Centre de Mathématiques Appliquées (CMAP), CNRS, Ecole Polytechnique, Institut Polytechnique de Paris, 91128 Palaiseau Cedex, France.

1 Introduction

This article aims at investigating the use of generative methods based on neural networks to simulate extreme events. Although very popular, these methods are mainly invoked in empirical works. Therefore, providing theoretical guidelines for using such models in extreme values context is of primal importance. To this end, we propose an overview of most recent generative methods dedicated to extremes, giving some theoretical and practical tips on their tail behaviour thanks to both extreme-value and copula tools. We begin by recalling the basic principles of generative modelling in Section 2 focusing on three particular methods: Generative adversarial networks (GANs), Variational autoencoders (VAEs) and Diffusion models. Section 3 provides a short description of copula tools to assess dependence between random variables. The limitations of classical GANs to deal with extreme events are highlighted in Section 4 on both simulated and real financial data. Section 5 describes the extreme-value framework used in Section 6 to interpret the new generative methods proposed in the literature dedicated to extremes. Some of these improvements are illustrated on simulated multivariate data in Section 7 and future directions are mentioned in Section 8.

2 Generative modeling

Generative models aim at mimicking the distribution of a random object, possibly in high dimension. These models are referred to as digital twins in industrial engineering, see [46, 51] for reviews. Generative models are particularly useful for instance in the field of data-augmentation: Enriching a dataset may reduce overfitting and improve the performance of statistical models. Another perspective of high interest is data-privacy: Sharing generated data with the same statistical properties as some confidential ones. In the context of generative modeling,

two different point of views may be distinguished. On the first hand, sampling complex motions of physical objects was originally done by solving equations describing the object behavior under constraints and by generating trajectories given different initial conditions. Such an approach requires to know the exact evolution formula or to build by hand the physical model using mathematical equations; This is refereed to as physics-based models. On the other hand, a new class of data-based generative models has recently emerged in the paradigm of artificial intelligence. Instead of looking for a physics-based model, one can try to learn it directly from the data using random noise as input (data-driven model). Such algorithms have the advantage of being fast in the simulation phase compared to their physical model counterparts, even though they may be slow in the inference phase. Thanks to the numerical and theoretical advances in the XXIst century, neural networks have proven to be excellent candidates as universal approximation functions. Among the neural network generative models developed so far [26], the most popular ones have been the Variational autoencoder (VAE) [43] based on variational inference, and the Generative adversarial network (GAN) [33], on which we shall focus on, based on a min-max game. More recent models such as Normalizing flows [56] and Diffusion models [59] have gained some popularity. Nowadays, both the construction and the optimization of heavy neural network models are made easy by open-source libraries (*e.g.* TensorFlow [1] or PyTorch [53]), resulting in an extraordinary interest of people coming from different communities and various mathematical backgrounds.

2.1 Theoretical framework

Let X be a random variable taking values in a general metric space $(\mathcal{X}, d_{\mathcal{X}})$ and let $\mathcal{P}(\mathcal{X})$ be the space of all probability measures defined on the Borel σ-algebra $\mathcal{B}_{\mathcal{X}}$. Then, given some observations $\{X_i \in \mathcal{X}\}_{i=1,\dots,n}$ drawn from an unknown distribution p_X on $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$, the objective in generative modeling is to find a function $G : \mathcal{Z} \to \mathcal{X}$, called a generator, and a probability distribution p_Z, called latent distribution and defined on some set $\mathcal{Z}$, such that

$$G(Z) \overset{d}{=} X, \text{ with } Z \sim p_Z. \tag{1}$$

The existence of G and p_Z is provided by the following Kuratowski's Theorem [11, Proposition 7.15], sometimes also called the measurable isomorphism Theorem [60, Page 7], in the case where $\mathcal{Z}$ and $\mathcal{X}$ are Polish spaces, *i.e.* complete and separable metric spaces.

Theorem 2.1 (Kuratowski)**.** *Let $(\mathcal{Z}, \mu_Z)$ and $(\mathcal{X}, \mu_X)$ two Polish probability spaces. Then there exists a (non-unique) measurable bijection $G : \mathcal{Z} \to \mathcal{X}$ such that $\mu_Z(G^{-1}(E)) = \mu_X(E)$ and $\mu_X(G(F)) = \mu_Z(F)$, for all Borel sets $E \subset \mathcal{X}$ and $F \subset \mathcal{Z}$.*

Here, we focus on a parametric family of generators $\mathcal{G} := \{G_\theta\}_{\theta \in \Theta}$ and we denote by $\mathcal{P} := \{p_\theta\}_{\theta \in \Theta}$ the set of associated parametric distributions such that

$G_\theta(Z) \sim p_\theta$. The problem comes to finding the best parameter $\theta^\star$ such that $p_{\theta^\star}$ and p_X are as close as possible, or equivalently

$$G_{\theta^\star}(Z) \overset{d}{\approx} X, \tag{2}$$

for a given $Z \sim p_Z$. A generative modeling problem mainly consists in choosing three ingredients:

1. The observations $X_1, \ldots, X_n$ with their underlying properties;
2. The parametrization G_θ and the latent distribution p_Z to use as inputs;
3. The distance or the similarity criterion between the probability distributions p_θ and p_X as well as the optimization process.

In the new paradigm of artificial intelligence, it is natural to consider a neural network as parametrization G_θ.

2.2 Neural networks

A neural network is a non-linear function built with a fixed number of neurons, each one representing a function, and distributed across several hidden layers. Neurons are scaled and translated in the network by parameters called respectively weights and biases. Among many different existing neural network architectures, let us consider the classical one-hidden layer feedforward neural network $G_{\theta_K} : \mathbb{R}^{d'} \to \mathbb{R}$ composed by K neurons such that

$$\mathbf{z} \in \mathbb{R}^{d'} \mapsto G_{\theta_K}(\mathbf{z}) = b^{(2)} + \sum_{k=1}^{K} w_k^{(2)} \sigma\left(\left\langle \mathbf{w}_k^{(1)}, \mathbf{z} \right\rangle + b_k^{(1)}\right) \in \mathbb{R}, \tag{3}$$

with parameter

$$\theta_K := \{b_2\} \cup \left\{\mathbf{w}_k^{(1)}, w_k^{(2)}, b_k^{(1)}\right\}_{k=1,\ldots,K} \in \Theta_K := \mathbb{R} \times (\mathbb{R}^{d'} \times \mathbb{R} \times \mathbb{R})^K,$$

and where $\langle \cdot, \cdot \rangle$ is a scalar product on $\mathbb{R}^{d'}$, and $\sigma : \mathbb{R} \to \mathbb{R}$ is a non-linear function called activation function (from now on, multivariate quantities are denoted by bold symbols). See Figure 1 for the illustration of a neural network with $K = 4$ neurons and $d' = 3$ as input dimension. Note that (3) can be interpreted as a particular case of the Ridge model [55] defined as

$$G(\mathbf{z}) = \sum_{k=1}^{K} g_k(\langle \mathbf{a}_k, \mathbf{z} \rangle),$$

with $g_k : \mathbb{R} \to \mathbb{R}$ and $\mathbf{a}_k \in \mathbb{R}^{d'}$ for all $k \in \{1, \ldots, K\}$. Examples of activation functions include:

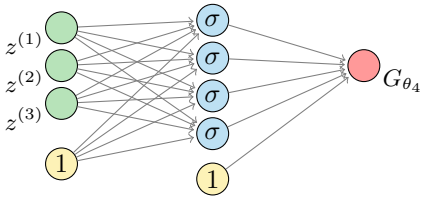

Figure 1: Example of a one-hidden layer neural network with $K = 4$ neurons and $d' = 3$.

- The cosine squasher

$$\sigma(x) = \frac{\cos(x + 3\pi/2) + 1}{2} \mathbb{1}\left\{x \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]\right\} + \mathbb{1}\left\{x \in \left(\frac{\pi}{2}, \infty\right)\right\}; \quad (4)$$

- The logistic squasher

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}}; \quad (5)$$

- The exponential Linear Unit (eLU) defined for all $\alpha > 0$ by

$$\sigma_\alpha(x) = \alpha(\exp(x) - 1)\mathbb{1}\{x < 0\} + x\mathbb{1}\{x \geq 0\}; \quad (6)$$

- The Rectified Linear Unit (ReLU)

$$\sigma(x) = \max(x, 0). \quad (7)$$

More generally, any increasing function σ such that $\sigma(x) \to 0$ as $x \to -\infty$ and $\sigma(x) \to 1$ as $x \to +\infty$ is called a sigmoidal squashing function. This definition includes (4) and (5) but excludes (6) and (7).

During the end of the XXth century, several authors studied the ability of G_{θ_K} defined in (3) to approximate a given function G as $K \to \infty$ under various assumptions on G and with respect to different norms. Let us summarize some of them:

- If G is a square integrable function on $[0, 1]^{d'}$, then [27] showed that there exists a "Fourier neural network" G_{θ_K} designed with cosine squasher activation functions (4) which converges in the L_2-sense to G as $K \to \infty$.

- If G is a continuous function on the d'-dimensional unit cube $[0, 1]^{d'}$, then [18] proved that there exists a neural network G_{θ_K} designed with sigmoidal squashing functions which converges uniformly to G as $K \to \infty$.

- The latter result is extended in [39] to functions G continuous on any compact set of $\mathbb{R}^{d'}$ and to bounded and non-constant activation functions. Moreover, [48] proved that the uniform convergence holds on compact sets if and only if the activation functions are not polynomial.

Nowadays, the reference result is due to [54] and is given below.

Theorem 2.2 (Universal approximation theorem)**.** *Suppose G is a continuous function on a compact space $\mathcal{Z} \subset \mathbb{R}^{d'}$ and σ is not a polynomial. Then, $\forall \varepsilon > 0$, there exists a neural network G_{θ_K} (for some K depending on ε) defined in* (3) *such that*

$$\sup_{\mathbf{z} \in \mathcal{Z}} |G(\mathbf{z}) - G_{\theta_K}(\mathbf{z})| < \varepsilon.$$

The key assumption of the Universal approximation theorem is that the function G to approximate is continuous on a compact set and thus bounded. We shall see in the following that this hypothesis is violated when dealing with heavy-tailed random variables X and the resulting practical consequences. Before that, we briefly describe in the next paragraphs the Generative Adversarial Networks and Variational Autoencoders that are convenient tools to estimate the parameter θ_K when the conditions of Theorem 2.2 are satisfied.

2.3 Generative Adversarial Network (GAN)

As proposed by [33], a GAN scheme aims at approximating the unknown generator $G : \mathbb{R}^{d'} \to \mathbb{R}^d$ through a parametric family of neural networks

$$\left\{ G_\theta : \mathbb{R}^{d'} \to \mathbb{R}^d,\, \theta \in \Theta \right\}, \tag{8}$$

and to estimate the optimal parameter $\theta^\star$ from a data set $\left\{ \mathbf{x}_i \in \mathbb{R}^d \right\}_{i=1,\dots,n}$ of independent samples from the unknown distribution p_X. The estimation is performed by optimizing an objective function which can be interpreted as an adversarial game between a generator in (8) and a discriminator chosen in a parametric family of functions

$$\{ D_\phi : \mathbb{R}^d \to [0,1],\, \phi \in \Phi \}.$$

In other words, $D_\phi(\mathbf{x})$ represents the probability that an observation $\mathbf{x} \in \mathbb{R}^d$ is drawn from $\mathbf{X} \sim p_X$. Both the generator and the discriminator are neural networks with opposite objectives: The former tries to mimic real data which seem likely by the discriminator, while the latter tries to distinguish between the two sources. See Figure 2 for an illustration. In [33], this optimization problem is defined as:

$$\arg\min_{\theta \in \Theta} \max_{\phi \in \Phi} \left[\mathbb{E}_{p_X} \left(\log D_\phi(\mathbf{X}) \right) + \mathbb{E}_{p_Z} \left(\log \left(1 - D_\phi \left(G_\theta(\mathbf{Z}) \right) \right) \right) \right]. \tag{9}$$

Statistical results on the estimators obtained by considering the empirical counterpart of the above optimization problem can be found in [13]. We also refer to [8] for the alternative Wasserstein GAN method and to [14, 35] for the associated theoretical properties.

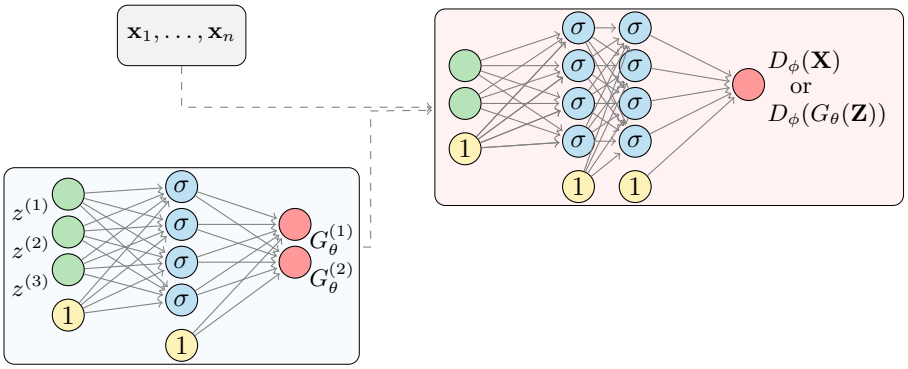

Figure 2: GAN model with $d' = 3$ and $d = 2$.

2.4 Variational Autoencoders (VAE)

The VAE approach introduced in [43] differs by considering two parametric families of mapping functions on $\mathcal{X}$ and $\mathcal{Z}$: the encoding set

$$\left\{\mathcal{C}_\phi : \mathbb{R}^d \to \mathbb{R}^{d'}, \phi \in \Phi\right\}$$

and the decoding set

$$\left\{\mathcal{D}_\theta : \mathbb{R}^{d'} \to \mathbb{R}^d, \theta \in \Theta\right\},$$

with their associated families of parametric densities $\left\{p^{\mathcal{C}_\phi}\right\}_{\phi\in\Phi}$ and $\left\{p^{\mathcal{D}_\theta}\right\}_{\theta\in\Theta}$. In order to achieve (1), the VAE setting aims at, for all $\mathbf{x} \in \mathcal{X}$:

1. maximizing the likelihood $p^{\mathcal{D}_\theta}(\mathbf{x}) = \int p^{\mathcal{D}_\theta}_{X\,|\,Z}(\mathbf{x}\,|\,\mathbf{z})p_Z(\mathbf{z})\,\mathrm{d}\mathbf{z}$ induced by the distribution of $\mathbf{Z}$ (which is usually known) in the decompression phase,
2. minimizing a distance or a divergence between $p^{\mathcal{C}_\phi}_{Z\,|\,X}(\cdot\,|\mathbf{x})$ and $p^{\mathcal{D}_\theta}_{Z\,|\,X}(\cdot\,|\mathbf{x})$.

Considering the KL-divergence

$$\mathtt{D}_{\mathtt{KL}}\left(p^{\mathcal{C}_\phi}_{Z\,|\,X}(\cdot\,|\,\mathbf{x}), p^{\mathcal{D}_\theta}_{Z\,|\,X}(\cdot\,|\,\mathbf{x})\right) = -\int_{\mathcal{Z}} p^{\mathcal{C}_\phi}_{Z\,|\,X}(\mathbf{z}\,|\,\mathbf{x}) \log\left(\frac{p^{\mathcal{D}_\theta}_{Z\,|\,X}(\mathbf{z}\,|\,\mathbf{x})}{p^{\mathcal{C}_\phi}_{Z\,|\,X}(\mathbf{z}\,|\,\mathbf{x})}\right)\mathrm{d}\mathbf{z}$$

in the above second objective and using the Bayes' rule, we get,

$$\mathcal{G}(\mathbf{x}) := \log p^{\mathcal{D}_\theta}(\mathbf{x}) - \mathtt{D}_{\mathtt{KL}}\left(p^{\mathcal{C}_\phi}_{Z\,|\,X}(\cdot\,|\,\mathbf{x}), p^{\mathcal{D}_\theta}_{Z\,|\,X}(\cdot\,|\,\mathbf{x})\right) \tag{10}$$

$$= \mathbb{E}_{p^{\mathcal{C}_\phi}_{Z\,|\,X}}\left(\log p^{\mathcal{D}_\theta}_{X\,|\,Z}(\mathbf{x}\,|\,\mathbf{Z})\right) - \mathtt{D}_{\mathtt{KL}}\left(p^{\mathcal{C}_\phi}_{Z\,|\,X}(\cdot\,|\,\mathbf{x}), p^{\mathcal{C}_\phi}_{Z}(\cdot)\right), \tag{11}$$

where $\mathcal{G}$ is called the Evidence Lower Bound.

It appears in (10) that $\mathcal{G}$ is indeed a lower bound for the log-likelihood $\log p^{\mathcal{D}_\theta}(\mathbf{x})$, the error term $\mathtt{D}_{\mathtt{KL}}(p^{\mathcal{C}_\phi}_{Z\,|\,X}(\cdot\,|\,\mathbf{x}), p^{\mathcal{D}_\theta}_{Z\,|\,X}(\cdot\,|\,\mathbf{x}))$ being small if $p^{\mathcal{C}_\phi}_{Z\,|\,X}$ is able to produce $\mathbf{z}$'s that can reproduce $\mathbf{x}$. The second representation of $\mathcal{G}$ in (11) is used for the numerical algorithm. The optimization program is:

$$\max_{\theta,\phi} \frac{1}{n}\sum_{i=1}^{n} \mathcal{G}(\mathbf{x}_i). \tag{12}$$

The optimization of (12) proposed in [43] mainly relies on Gaussian multivariate densities with diagonal covariance matrices:

$$\begin{aligned} p_Z(\cdot) &\overset{d}{=} \mathcal{N}(\mathbf{0}, I_{\mathrm{d}'}), \\ p^{\mathcal{C}_\phi}_{Z\,|\,X}(\cdot\,|\,\mathbf{x}) &\overset{d}{=} \mathcal{N}(\boldsymbol{\mu}^{\mathcal{C}_\phi}(\mathbf{x}), \mathtt{diag}\left(\Sigma^{\mathcal{C}_\phi}(\mathbf{x})\right)) \\ &\overset{d}{=} \boldsymbol{\mu}^{\mathcal{C}_\phi}(\mathbf{x}) + \left(\mathtt{diag}\left(\Sigma^{\mathcal{C}_\phi}(\mathbf{x})\right)\right)^{1/2} \mathcal{N}(\mathbf{0}, I_{\mathrm{d}'}), \end{aligned}$$

where $\left\{\boldsymbol{\mu}^{\mathcal{C}_\phi}(\mathbf{x}), \Sigma^{\mathcal{C}_\phi}(\mathbf{x})\right\}_{\phi\in\Phi}$ are parametrized by neural networks in order to have a closed form of the KL-divergence and a tractable gradient in (11). The parametrization of $p^{\mathcal{D}_\theta}_{X\,|\,Z}(\mathbf{x}\,|\,\cdot)$ is either Gaussian or Bernoulli whether the data are respectively continuous or discrete. See Figure 3 for an illustration and [22, 42] for more details on VAEs.

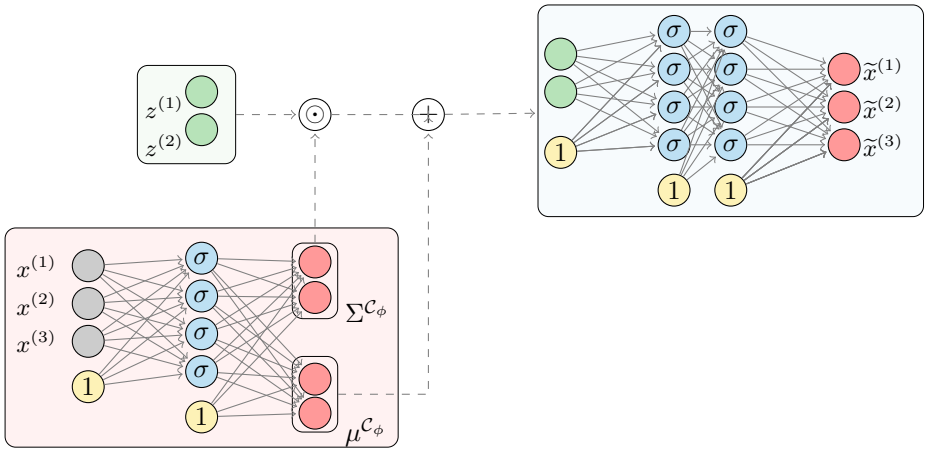

Figure 3: VAE model with $d' = 2$ and $d = 3$. The operator $\odot$ denotes the component-wise product and $\widetilde{x}^{(j)}$ corresponds to the j-th component of the generated data vector.

2.5 Diffusion models

In the context of generative modelling, these models date back to the work of [59], see also [37]. It can be viewed as a decomposition of the Encoder and

Decoder stages of VAE using infinitely many infinitesimal transformations, that are performed by running a continuous-time *forward* and *backward* Stochastic Differential Equation (a.k.a. diffusion equation).

The method is based on the following reverse-time result by [7]. Consider a d-dimensional diffusion process $\mathbf{X}$ driven by a Brownian motion $\mathbf{W}$, with a drift coefficient $\mathbf{b}(\cdot)$:

$$\mathbf{X}_t = \mathbf{X}_0 + \int_0^t \mathbf{b}(\mathbf{X}_s)\,\mathrm{d}s + \mathbf{W}_t, \quad \mathbf{X}_0 \sim p_0(\cdot). \tag{13}$$

Denote by $p_t(\cdot)$ the distribution density of $\mathbf{X}_t$; At time $t = 0$, this is the data distribution. Then $(\mathbf{Y}_t = \mathbf{X}_{T-t} : 0 \leq t \leq T)$ has the same law as the solution to the diffusion equation

$$\mathbf{Y}_t = \mathbf{Y}_0 + \int_0^t \Big(-\mathbf{b}(\mathbf{Y}_s) + \partial_y \log p_{T-s}(\mathbf{Y}_s) \Big)\,\mathrm{d}s + \mathbf{W}_t, \quad \mathbf{Y}_0 \sim p_T(\cdot). \tag{14}$$

Running $\mathbf{X}$ from $t = 0$ to $t = T$ (forward path) corresponds to the Encoder, yielding a latent distribution $p_T(\cdot)$, while running $\mathbf{Y}$ from $t = 0$ to $t = T$ (backward path) stands for the Decoder. The so-called score $\partial_y \log p_{T-s}(\cdot)$ is approximated by Neural networks. The choice of the latent distribution is quite flexible (a Gaussian distribution for example) since the drift $\mathbf{b}(\cdot)$ is a free-parameter that can be chosen (Langevin scheme) so that in large time T, p_T is close to a prescribed latent distribution; see [19] for detailed discussions.

So far, to the best of our knowledge, these diffusion models have not been used to generate extreme values.

3 Copulas

Let us consider a cumulative distribution function F_X defined on $\mathbb{R}^d$ with continuous margins denoted by $F_X^{(j)}$, $j \in \{1, \ldots, d\}$. From Sklar's Theorem [58], there exists a unique function $C : [0,1]^d \to [0,1]$ such that

$$F_X(\mathbf{x}) = C\left(F_X^{(1)}(x^{(1)}), \ldots, F_X^{(d)}(x^{(d)})\right),$$

for all $\mathbf{x} = \left(x^{(1)}, \ldots, x^{(d)}\right) \in \mathbb{R}^d$. The function C is called the copula of F_X. Introducing the uniform random variables $U^{(j)} = F^{(j)}(X^{(j)})$ for all $j \in \{1, \ldots, d\}$, the copula C is the $d-$ dimensional distribution function of the random vector $\left(U^{(1)}, \ldots, U^{(d)}\right)$ defined on the unit cube $[0,1]^d$ with uniform margins on $[0,1]$. Copulas are a flexible tool to impose a given dependence structure on the marginal distributions of interest, see [52] for a detailed account on copulas. The independence between margins corresponds to the product copula $\Pi(\mathbf{u}) = u^{(1)} \ldots u^{(d)}$ while comotonic dependence corresponds to the Fréchet copula $M(\mathbf{u}) = \min(u^{(1)}, \ldots, u^{(d)})$.

3.1 Archimedean copulas

An Archimedean copula C_μ is defined for all $\mathbf{u} = (u^{(1)}, \ldots, u^{(d)}) \in [0,1]^d$ by

$$C_\mu(\mathbf{u}) = \psi_\mu\left(\psi_\mu^{-1}(u^{(1)}) + \cdots + \psi_\mu^{-1}(u^{(d)})\right),$$

where $\psi_\mu : [0,\infty) \to [0,1]$ is a parametric function called generator which has to verify some properties listed for instance in [49]. It can easily be seen that the independence copula Π is Archimedean, generated by $\psi(t) = \exp(-t)$, $t \geq 0$. In the following, we shall focus on the Gumbel copula,

$$C_\mu^{\mathrm{G}}(\mathbf{u}) = \exp\left(-\left(\sum_{j=1}^{d}(-\log u_j)^\mu\right)^{1/\mu}\right),$$

where $\mu \geq 1$ tunes the dependence between the margins, see the next paragraph. This copula has been proved to be the only max-stable Archimedean copula [30]. The associated generator is $\psi_{C_\mu^{\mathrm{G}}}(t) = \exp(-t^{1/\mu})$ defined for all $\mu \geq 1$ and $t \geq 0$.

3.2 Quantifying the dependence

First, Kendall's dependence function [31] characterizes the dependence structure associated with a copula C and is the univariate cumulative distribution function defined by $K_C(t) = \mathbb{P}\left(C\left(U^{(1)}, \ldots, U^{(d)}\right) \leq t\right)$ for all $t \in [0,1]$. In the case of an Archimedean copula C_μ, it can be derived as [28, Equation (6)]:

$$K_{C_\mu}(t) = t + \sum_{j=1}^{d-1} \frac{(-\psi_\mu^{-1}(t))^j}{j!}\psi_\mu^{(j)}(\psi_\mu^{-1}(t)),$$

where $\psi_\mu^{(j)}$ stands for the j-th derivative of ψ_μ, and we shall thus consider $\lambda_{C_\mu}(t) := t - K_{C_\mu}(t)$. It is then easily seen that $\lambda_M(t) = 0$ and

$$\lambda_\Pi(t) = -t\sum_{j=1}^{d-1}\frac{(-\log(t))^j}{j!} \tag{15}$$

for all $t \in (0,1]$. Let us also highlight that $\lambda_\Pi(t) \to \lambda_{\Pi,\infty}(t) := t - 1$ as $d \to \infty$. Moreover, Kendall's dependence function associated with the Gumbel copula is given in the bivariate case by $K_{C_\mu^{\mathrm{G}}}(t) = t - t\log(t)/\mu$ leading to $\lambda_{C_\mu^{\mathrm{G}}}(t) = t\log(t)/\mu$ for all $t \in (0,1]$. Clearly, $\lambda_{C_1^{\mathrm{G}}}(t) = \lambda_\Pi(t) = t\log(t)$ and $\lambda_{C_\mu^{\mathrm{G}}}(t) \to \lambda_M(t) = 0$ as $\mu \to \infty$.

Second, Kendall's tau [41] is a measure of dependence between two random variables. Let us then assume $d = 2$ and let $\mathbf{X}$ and $\tilde{\mathbf{X}}$ be two bivariate random vectors from F_X. Kendall's tau is defined as the probability of concordance minus the probability of discordance of $\mathbf{X} = (X^{(1)}, X^{(2)})$ and $\tilde{\mathbf{X}} = (\tilde{X}^{(1)}, \tilde{X}^{(2)})$.

From [52, Theorem 5.1.3], this quantity only depends on the copula C of F_X and is given by

$$\tau_C = 4\mathbb{E}\Big(C(U^{(1)}, U^{(2)})\Big) - 1 = 4\int_0^1 \int_0^1 C(u,v)\,\mathrm{d}C(u,v) - 1, \tag{16}$$

with $\tau_M = 1$ and $\tau_\Pi = 0$ as special cases. In case of an Archimedean copula C_μ, Kendall's tau and Kendall's dependence functions are linked [29]:

$$\tau_{C_\mu} = 1 + 4\int_0^1 \lambda_{C_\mu}(v)\,\mathrm{d}v,$$

meaning that τ_{C_μ} can be interpreted as a summary of the dependence information encoded in $\lambda_{C_\mu}(\cdot)$. As an example, the Kendall's tau associated with the Gumbel copula is given for all $\mu \geq 1$ by $\tau_{C_\mu^{\mathrm{G}}} = 1 - 1/\mu$. The dependence between the margins is thus an increasing function of μ. Note in particular that $\tau_{C_1^{\mathrm{G}}} = \tau_\Pi = 0$ while $\tau_{C_\mu^{\mathrm{G}}} \to \tau_M = 1$ as $\mu \to \infty$.

3.3 Sampling (bivariate case)

Sampling a random pair (U, V) from a bivariate copula C can be achieved by first simulating independently $(U, W) \sim \mathcal{U}([0,1]^2)$ and then letting $V = C_u^{-1}(W)$ where C_u is the conditional copula defined by

$$C_u(v) = \mathbb{P}(V \leq v | U = u) = \partial_u C(u, v).$$

In the case of bivariate Archimedean copulas, the conditional copula and its inverse are given by [10]:

$$C_{\mu,u}(v) = \frac{\partial_u\left(\psi_\mu^{-1}\right)(u)}{\partial_u\left(\psi_\mu^{-1}\right)(C(u,v))},$$
$$C_{\mu,u}^{-1}(y) = \psi_\mu\left(\left(\partial_u\psi_\mu\right)^{-1}\left(\frac{y}{\partial_u\left(\psi_\mu^{-1}\right)(u)}\right) - \psi_\mu^{-1}(u)\right).$$

We also refer to [38, 63] for alternative methods based on Laplace transform and Kendall's dependence function respectively.

3.4 Inference

The estimation of Kendall's dependence function is based on the pseudo-observations $\{W_1, \ldots, W_n\}$ from the cumulative distribution function K and computed as

$$W_i = \frac{1}{n-1}\sum_{j\neq i}^{n} \mathbb{1}\left\{X_j^{(1)} < X_i^{(1)}, \ldots, X_j^{(d)} < X_i^{(d)}\right\}, \tag{17}$$

for all $i \in \{1, \ldots, n\}$, see [31]. The estimator of K is computed using the associated empirical cumulative distribution function:

$$\hat{K}_n(t) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{W_i \leq t\},$$

and we set $\hat{\lambda}_n(t) = t - \hat{K}_n(t)$, for all $t \in [0, 1]$. Similarly, see [31, eq. (7)], Kendall's tau is estimated by

$$\hat{\tau}_n = \frac{4}{n} \sum_{i=1}^{n} W_i - 1.$$

4 Simulating extremes with GANs, numerical illustrations

The GAN ability to simulate extreme observations in heavy tails is assessed on two situations: Real financial data (Paragraph 4.3) and simulated bivariate data from a Gumbel copula and marginal Burr distributions (Paragraph 4.4). We begin by providing some implementation details in the next two paragraphs.

4.1 Performance assessment

Let us denote by $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n\}$ the outputs generated by the GAN model, where $\tilde{\mathbf{x}}_i = G_{\theta^\star}(\mathbf{z}_i)$, $i \in \{1, \ldots, n\}$ and with $\tilde{\mathbf{x}}_{1,n} \leq \cdots \leq \tilde{\mathbf{x}}_{n,n}$ the associated order statistics. The fit on the marginal distribution tails is assessed using the Mean squared logarithmic error (MSLE) defined as

$$\mathrm{MSLE}(\xi) = \frac{1}{d\lceil (1-\xi)n \rceil} \sum_{j=1}^{d} \sum_{i=1}^{\lceil (1-\xi)n \rceil} \left(\frac{\log(x^{(j)}_{n-i+1,n}) - \log(\tilde{x}^{(j)}_{n-i+1,n})}{\log(2)} \right)^2.$$

In the sequel, we use $\xi \in \{0.90, 0.95, 0.99\}$ to focus the computation on the tails. Note that the $\log(2)$ factor is introduced so that a 100% relative error on the log-marginals yields $\mathrm{MSLE}(\xi) = 1$. Considering the dependence structure, one may graphically compare the estimated Kendall's dependence functions K (or equivalently the $t \mapsto \lambda(t) := t - K(t)$ functions) on the n observations associated with the original and generated samples. We shall also compare the estimated Kendall's tau $\hat{\tau}_n$ and $\tilde{\tau}_n$ on the original and GAN samples.

4.2 Computational aspects

The Adam optimizer [21] is used with default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for all tests performed during $1,000$ iterations. Every 5 iterations, the MSLE metric is computed and the parameters of the ReLU neural

network associated with the best results among the 200 checkpoints are selected. The experiments have been conducted on the Cholesky computing cluster from Ecole Polytechnique `http://meso-ipp.gitlab.labos.polytechnique.fr/user_doc/`. It is composed by 2 nodes, where each one includes 2 CPU Intel Xeon Gold 6230 @ 2.1GHz, 20 cores and 4 Nvidia Tesla v100 graphics card. The code was implemented in Python 3.8.2 and using the library PyTorch 1.7.1 for the GANs' training.

4.3 Real financial data

The GAN approach is tested on closing prices of daily financial stock market indices downloaded from `https://stooq.com/db/h/` on the October 1st, 2020. Six indices are selected NKX (Nikkei, Japan), KOSPI (Korea), HSI (Hong-Kong), CAC (France), AMX (Amsterdam Exchange, Netherlands), Nasdaq (USA) from three market zones: Asia, Europe, USA. As a pre-processing step, the daily log-returns are computed for each ticker index and positive values are discarded to focus on the simulation of losses. In case of missing data at a given business day, the next available day is removed from the dataset. Besides, we kept only the data available at the same date for all selected tickers. The performance of the GAN approach is assessed on four datasets of increasing dimensions: NKX ($d = 1$), Europe (AEX, CAC, $d = 2$), Asia (NKX, KOSPI, HSI, $d = 3$) and world (AEX, CAC, NKX, KOSPI, HSI, NDQ, $d = 6$), see Table 1.

Table 1: MSLE(ξ) associated with the GAN approach on real financial data.

	NKX	Europe	Asia	World
Dimension d	1	2	3	6
Sample size n	3173	2504	1378	548
$\xi = 0.90$	0.984	8.034	4.897	6.881
$\xi = 0.95$	1.544	10.251	3.082	9.297
$\xi = 0.99$	2.874	5.811	2.129	10.407

It clearly appears that the GAN method is unable to reproduce the marginal distribution tails: For all quantile levels $\xi \in \{0.90, 0.95, 0.99\}$ and all considered datasets, the numerical results point towards relative errors close to or larger than 100%. The quality of the results even deteriorates when ξ or the dimension d increases.

These disappointing results can be graphically interpreted on Figure 4 where the performance of the generator is visually assessed by comparing log quantile-quantile plots, namely the pairs $(\log((n+1)/i), \log x^{(j)}_{n-i+1,n})$ and $(\log((n+1)/i), \log \tilde{x}^{(j)}_{n-i+1,n})$ for $i \in \{1, \ldots, \lceil (1-\xi)n \rceil\}$ and $j \in \{1, \ldots, 6\}$. The quantile-quantile plots computed on all indices at level $\xi = 0.95$ are approximately linear which provides a graphical evidence of the tail heaviness of all six marginal distributions, see Section 5 for theoretical details. It is apparent that GAN

samples do not reproduce well the heavy tail property of the original samples. This under-estimation of the tail heaviness can be quantified by computing the slopes associated with all quantile-quantile plots, thus providing an estimation of the tail index on all six datasets, see Table 2. Clearly, all GAN samples have much lighter tails than the original ones.

Table 2: Estimated tail indices associated with the GAN approach on real financial data.

Ticker	Original data	GAN data
AEX	0.268	0.124
CAC	0.292	0.135
NKX	0.357	0.114
KOPSI	0.251	0.120
HSI	0.226	0.127
NDQ	0.352	0.166

4.4 Simulated bivariate data

In this experiment, we consider simulated data from a Gumbel copula C_μ^{G} and with $\mathrm{Burr}(\gamma, \rho)$ margins. Its cumulative distribution function is given for all $x \geq 0$ by

$$F_{\mathrm{Burr}}(x) = 1 - (1 + x^{-\rho/\gamma})^{1/\rho}, \tag{18}$$

with $\gamma > 0$ and $\rho < 0$. Recall that the Burr distribution is heavy-tailed in the sense that it belongs to the Fréchet maximum domain of attraction [20, Theorem 1.2.1], with tail-index γ and second-order parameter ρ, see Paragraph 5.2.1 for theoretical details. A sample of size $n = 10,000$ is simulated from the random vector $\mathbf{X} = (X^{(1)}, X^{(2)})$ following the procedure described at Paragraph 3.3 with margins $X^{(1)} \sim \mathrm{Burr}(\gamma, \rho_1)$ and $X^{(2)} \sim \mathrm{Burr}(\gamma, \rho_2)$ linked by C_μ^{G} for different combinations of $(\mu, \gamma, \rho_1, \rho_2)$ in $\{1.1, 2, 10\} \times \{0.1, 0.5, 0.9\} \times \{-1, -2, -3\}^2$. The obtained values of $\mathrm{MSLE}(\xi = 0.99)$ are reported in Table 3. It appears that, when the tail index is large ($\gamma = 0.9$), all MSLE values are close to, or even larger than 1 which corresponds to a relative error of 100%. This simulated experiment confirms that GANs cannot reproduce heavy-tailed phenomena. This vexing property does not seem to depend neither on the second-order parameters nor on the dependence parameter. In contrast, Figure 5 and Table 4 show that the dependence structure is correctly reproduced in this low-dimensional situation (recall that $d = 2$): Estimated Kendall's tau and dependence functions on GAN and original data are close to each other, and very similar to the theoretical ones. We refer to Section 7 for experiments in higher dimensional settings.

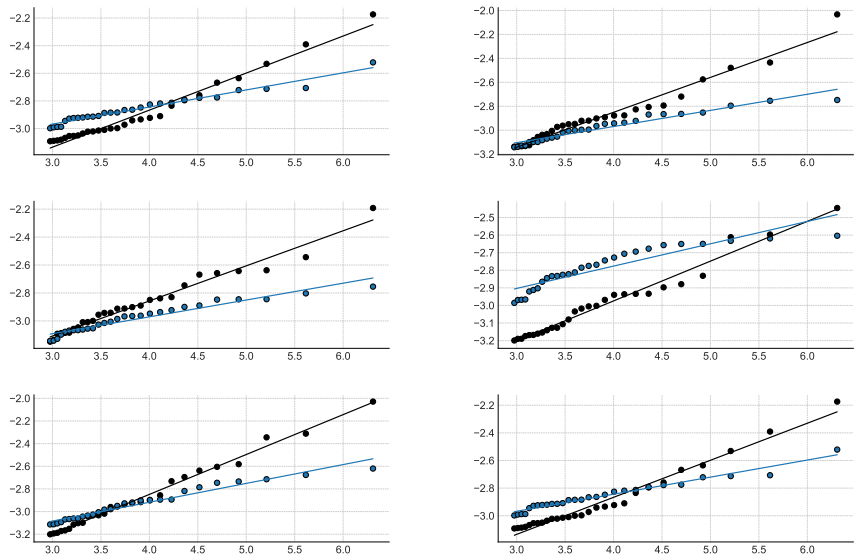

Figure 4: Quantile-quantile plots $\log((n+1)/i) \mapsto \log x^{(j)}_{n-i+1,n}$, for $i \in \{1,\ldots,\lceil(1-\xi)n\rceil\}$ and $j \in \{1,\ldots,6\}$ associated with six financial indices at probability level $\xi = 0.95$ (black: real data, blue: GAN data). From left to right and top to bottom: AEX, CAC, NKX, KOSPI, HSI and NDQ. The regression line is superimposed to each plot.

Table 3: MSLE($\xi = 0.99$) associated with the GAN approach on simulated bivariate data.

Tail-index	2nd-order parameters	Theoretical Kendall's tau		
γ	(ρ_1, ρ_2)	$\tau_{C^G_{1,1}} = 0.1$	$\tau_{C^G_2} = 0.5$	$\tau_{C^G_{10}} = 0.9$
0.1	$(-1,-2)$	0.019	0.011	0.005
	$(-1,-3)$	0.019	0.015	0.012
	$(-2,-3)$	0.014	0.017	0.015
0.5	$(-1,-2)$	0.074	0.220	0.053
	$(-1,-3)$	0.040	0.144	0.079
	$(-2,-3)$	0.225	0.209	0.027
0.9	$(-1,-2)$	0.994	1.152	1.190
	$(-1,-3)$	0.936	1.068	0.955
	$(-2,-3)$	1.424	0.756	0.933

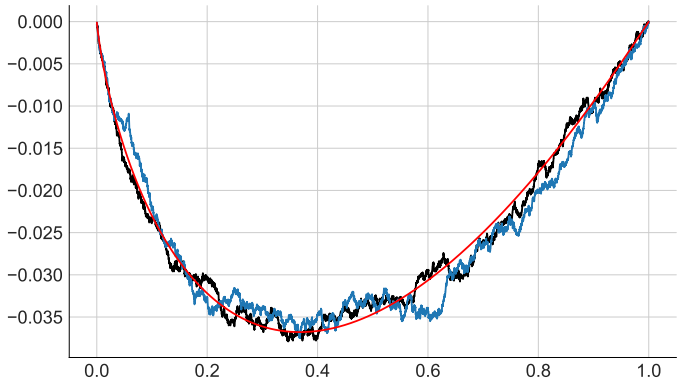

Figure 5: Estimated $t \in [0,1] \mapsto \lambda(t)$ functions. Black: original simulated data ($\gamma = 0.9$, $\rho_1 = -1$, $\rho_2 = -3$ and $\mu = 10$), blue: data generated with the GAN model. The theoretical $\lambda_{C_{10}^{\mathrm{G}}}$ function is superimposed in red.

5 Extreme-value framework

Let us first consider in Paragraph 5.1-5.3 a real random variable X with cumulative distribution function F_X defined on $\mathbb{R}$. In this one-dimensional setting, problem (1) benefits from an explicit solution based on the quantile function $u \in (0,1) \mapsto q_X(u) := \inf\{x : F_X(x) \geq u\}$. The inversion method [23] shows that one can set $G := q_X$ and $Z \sim \mathcal{U}([0,1])$. The multivariate setting is presented in Paragraph 5.4.

5.1 Heavy-tailed (univariate) distributions

We focus on heavy-tailed distributions, *i.e.* when F_X belongs to the Fréchet maximum domain of attraction [20, Theorem 1.2.1]. From [15], the survival function $\bar{F}_X := 1 - F_X$ can be written for x large enough as

$$\bar{F}_X(x) = x^{-1/\gamma} \ell_X(x), \tag{19}$$

where ℓ_X is a slowly-varying function at infinity: $\ell_X(\lambda x)/\ell_X(x) \to 1$ as $x \to \infty$ for all $\lambda > 0$. In such a case, $\bar{F}_X$ is said to be regularly-varying with index $-1/\gamma$ at infinity, which is denoted for short by $\bar{F}_X \in \mathcal{RV}_{-1/\gamma}$. Similarly, we shall note $\ell_X \in \mathcal{RV}_0$. The tail-index γ tunes the tail heaviness of the cumulative distribution function F_X: The larger γ is, the heavier the tail.

As a consequence of (19), the tail quantile function $x \mapsto q_X(1 - 1/x)$ is regularly-varying with index γ at infinity, see [20, Proposition B.1.9.9], or, equiv-

alently,

$$\log q_X(u) = \gamma \log\left(\frac{1}{1-u}\right) + \log L\left(\frac{1}{1-u}\right), \tag{20}$$

for all $u \in (0,1)$ with $L \in \mathcal{RV}_0$. Now, Since L is slowly-varying, $\log L(v)/\log v \to 0$ as $v \to \infty$ from [15, Proposition 1.3.6] and then,

$$\log q_X(u) = \gamma \log\left(\frac{1}{1-u}\right)(1+o(1)), \text{ as } u \to 1.$$

Note that the linearity of the log-quantile-quantile plots (Figure 4) is a consequence of the above property. Besides, $q_X(u) \to \infty$ as $u \to 1$ so that q_X does not fulfill the assumptions of Theorem 2.2: There is no theoretical guaranty that a neural network (3) could uniformly approximate G. Moreover, since Z is a bounded random variable, when the activation function σ is continuous, $G_{\theta_K}(Z)$ is also a bounded random variable and thus cannot be heavy-tailed. The disappointing behavior of the GAN observed in the previous two paragraphs can thus be explained in the light of these remarks.

In the following, we shall use an additional assumptions on F_X, or equivalently on L, to refine the heavy-tail model (19). To this end, consider the Karamata representation [20, Definition B1.6]:

$$L(x) = c(x) \exp\left(\int_1^x \frac{\varepsilon(t)}{t}\,\mathrm{d}t\right), \tag{21}$$

where $c(x) \to c_\infty$ as $x \to \infty$ and ε is a measurable function such that $\varepsilon(x) \to 0$ as $x \to \infty$. Our second main assumption then writes:

$$c(x) = c_\infty > 0 \text{ for all } x \geq 1 \text{ and } \varepsilon \in \mathcal{RV}_\rho \text{ with } \rho < 0. \tag{22}$$

The assumption that c is a constant function is equivalent to assuming that L is normalized [44] and ensures that L and thus q_X are differentiable. The condition $\varepsilon \in \mathcal{RV}_\rho$ with $\rho < 0$ entails that $L(x) \to c_\infty$ as $x \to \infty$. The index of regular variation ρ is referred to as the second-order parameter. It is the main driver of the bias in the estimation of extreme quantiles from heavy-tailed distributions. Besides, (22) implies that F_X satisfies the so-called second-order condition [20, Equation (2.3.22)] which is the cornerstone of all proofs of asymptotic normality in extreme-value statistics.

5.2 Examples of heavy-tailed distributions

The following three examples are repeatedly used in this work.

5.2.1 Burr distribution

Comparing (18) and (19), it is clear that the $\mathrm{Burr}(\gamma, \rho)$ distribution is heavy-tailed with slowly-varying function $\ell_X(x) = (1+x^{\rho/\gamma})^{1/\rho}$, $x \geq 0$. The associated

log-quantile function is given for all $u \in (0,1)$ by

$$\log q_X(u) = -\frac{\gamma}{\rho} \log\left((1-u)^\rho - 1\right),$$

so that $L(x) = (1-x^\rho)^{-\gamma/\rho}$, $x > 0$ and (22) holds with $c_\infty = 1$ and $\varepsilon(t) = \gamma/(t^{-\rho}-1)$. As illustrated in the top panel of Figure 6, the rate of divergence of $u \mapsto \log q_X(u)$ in the neighbourhood of $u = 1$ is mainly driven by the tail-index γ.

5.2.2 Generalized Pareto Distribution (GPD)

The cumulative distribution function of the GPD(γ, σ) is given by

$$F_{\text{GPD}}(x) = 1 - (1 + x/\sigma)^{-1/\gamma}, \tag{23}$$

for all $x \geq 0$, with $\gamma > 0$ and $\sigma > 0$. This definition can be extended to $\gamma \in \mathbb{R}$ without difficulty, but we limit ourselves to $\gamma > 0$ to ensure that the GPD is heavy-tailed. We shall also focus on the standard GPD with $\sigma = 1$ so that the GPD$(\gamma, \sigma = 1)$ coincides with the Burr$(\gamma, \rho = -\gamma)$, see (18) and (23).

5.2.3 Extreme Value Distribution (EVD)

The cumulative distribution function of the EVD(γ, σ, μ) is given for all $x \geq (\mu - \sigma)/\gamma$ by

$$F_{\text{EVD}}(x) = \exp\left(-(1 + \gamma(x-\mu)/\sigma)^{-1/\gamma}\right), \tag{24}$$

with $\gamma > 0$, $\sigma > 0$ and $\mu \in \mathbb{R}$. This definition can be extended to $\gamma \in \mathbb{R}$ and then gives rise to the limiting distribution for maxima of independent and identically distributed random variables. Here, we limit ourselves to $\gamma > 0$ to ensure that the EVD is heavy-tailed.

5.3 Risk measures

The Value-at-Risk (VaR) at level $\alpha \in (0,1)$ is the upper α-quantile of the loss distribution associated with the random variable X: $\text{VaR}_X(\alpha) = q_X(\alpha)$. The VaR suffers from several weaknesses: It is not a coherent risk measure [2] since it is not subadditive in general. A a consequence, it has been proposed [2, 57] to switch from the VaR to the Expected Shortfall (ES) defined as the average of the quantile function above a given confidence level $\alpha \in (0,1)$:

$$\text{ES}_X(\alpha) = \frac{1}{1-\alpha} \int_\alpha^1 q_X(u)\,\mathrm{d}u. \tag{25}$$

Unlike the VaR, it can be shown that the ES satisfies all of the requirements to be a coherent risk measure, namely translation invariance, monotonicity, positive

homogeneity, and subadditivity [9]. Note that the ES is a particular case of conditional tail moment [50] defined by

$$\mathrm{CTM}_{X,p}(\alpha) = \frac{1}{1-\alpha}\int_\alpha^1 q_X^p(u)\,\mathrm{d}u.$$

where $p > 0$.

5.4 Multivariate extremes

Let us consider the case where $\mathbf{X}$ is a random vector on $\mathbb{R}_+^d$. It can be decomposed into a radial component $R = \|\mathbf{X}\|_1 = X_1 + \cdots + X_d$ and an angular component on the $(d-1)$-dimensional simplex $\boldsymbol{\omega} = \mathbf{X}/\|\mathbf{X}\|$. Then, $\mathbf{X}$ is said to have multivariate regular variation if the following two properties hold:

- The survival function of R is regularly-varying as defined in (19);
- There exists a probability measure S defined on the $(d-1)$-dimensional simplex such that
$$\mathbb{P}\left(\boldsymbol{\omega} \in \cdot \mid R > r\right) \xrightarrow{w} S(\cdot), \tag{26}$$
where $\xrightarrow{w}$ denotes the weak convergence.

The limiting probability measure S is called the angular measure, it characterizes the dependence structure of multivariate extremes. The estimation of S is thus of primary interest for the analysis of multivariate extremes, see among others [24].

6 Adapting generative methods to extremes

It was from 2020 onwards that the scientific community became aware of the need to adapt generative methods to extremes, and most of the ensuing works were dedicated to GANs, see Paragraph 6.1 for an overview and Paragraph 6.2 for other architectures.

6.1 Improvements of GANs

Three main directions have been investigated to adapt GANs to heavy tails: A preprocessing of the data to get rid of the tail heaviness, the use of heavy-tailed latent variables, and the introduction of new parametrizations to adapt the optimization problem (9) to the heavy-tail situation.

6.1.1 Preprocessing: Quant-GAN and evtGAN

Let us also emphasize that, if the latent variable Z is chosen to be Gaussian, then piecewise linear transforms of Z obtained with (3) combined with ReLU functions (7) remain Weibull-tailed [61] and therefore cannot be heavy-tailed.

In [62], it is remarked that log-returns of some financial indices are well represented by Lambert W transforms of Gaussian random variables. Following the ideas of [32], they thus propose to use an inverse Lambert W transform to gaussianize the data. This preprocessing step is part of the Quant-GAN methodology introduced in [62]. The generator outputs are finally transformed back using the direct Lambert function $W(x) = x\exp(\gamma x^2/2)$ for recovering the heavy-tailed data property. Here $\gamma > 0$ is the tail-index as defined in (19) which has to be estimated outside the GAN methodology. Similarly, in evtGAN [16], extreme-value distributions (see Paragraph 5.2.3) are fitted to the margins, which are then transformed to uniform random variables. A classical GAN is then applied before transforming back the margins of the simulated samples using the fitted extreme-value distributions.

6.1.2 New latent variables: Pareto-GAN

Another, but similar, approach is to use directly a heavy-tailed latent variable in the GAN setting [25, 40]. It is proposed in [40] to use a GPD, see Paragraph 5.2.2. It is then shown that the generator outputs follow the desired heavy-tailed distribution: A piecewise linear transform of the above GPD has still tail-index γ. Alternative metric spaces are also introduced to ensure the loss function to be finite. To be effective, the so-called Pareto-GAN method, similarly to Quant-GAN, requires the accurate estimation of the tail-index associated with each heavy-tailed marginal distribution. This is a challenging task in extreme-value theory, see [20, Chapter 3].

6.1.3 New parametrizations: EV-GAN and Tail-GAN

In [4], it is proposed to take advantage of the quantile representation (20) established in the heavy-tail framework (19) to introduce a new parametrization of GANs. More specifically, it is remarked that the so-called tail-index function

$$u \in [0,1) \mapsto f^{\mathrm{TIF}}(u) := -\frac{\log q_X(u)}{\log\left((1-u^2)/2\right)}, \tag{27}$$

is continuous, tends to the tail-index γ as $u \to 1$ and is thus bounded on $[0,1]$, see the middle panel of Figure 6 for an illustration in the Burr case. As such, f^{TIF} fulfils the assumptions of Theorem 2.2 and can thus be uniformly approximated by a neural network. Under the additional assumption (22), a corrected version f^{CTIF} of the tail-index function is introduced. It is then shown [4, Proposition 1] that, if $\rho < -1$, then f^{CTIF} is continuously differentiable on $[0,1]$, and the approximation error in Theorem 2.2 is derived as a function of K, the number of neurons in (3). We refer to [4, Theorem 4] for technical details and other ranges of ρ values.

In [17], the joint elicitability property [3] of the VaR and ES risk measures (see Paragraph 5.3) is exploited to propose a new GAN parametrization. A universal approximation theorem is provided for a broad class of tail risk mea-

sures: Any Hölder continuous spectral risk measure can be approximated with an arbitrary precision by the proposed GAN architecture.

6.1.4 Heuristics: Ex-GAN

Alternatively, in [12], a distribution shifting is first introduced in order to reduce the lack of training data in the tails. Second, a GAN parametrization conditioned by samples drawn from a GPD (see Paragraph 5.2.2) is fitted to the shifted data. Finally, an additional term representing some distance to a desired extremeness is added to the loss function. Although numerical results on images are promising, we do not think that the proposed parametrization gives theoretical support for generating extreme observations in the sense that no error or complexity bounds are provided in the NN architecture of the generator.

6.2 Other architectures

We finally list two works outside the GAN framework and dedicated to the simulation of multivariate extremes.

6.2.1 Improvements of VAEs

It is shown in [47, Corollary 7] that a VAE built with piecewise linear activation functions and Gaussian distributions for both $p^{\mathcal{C}_\phi}_{Z\,|\,X}(\cdot\,|\,\mathbf{x})$ and $p^{\mathcal{D}_\theta}_{X\,|\,Z}(\mathbf{x}\,|\,\cdot)$ cannot reproduce heavy-tailed margins. It is also proved, under some assumptions, that the angular measure (26) associated with a classical VAE output is necessarily discrete, see [47, Proposition 8]. To overcome these problems, the authors consider an univariate heavy-tailed distribution to sample the radius R, and, conditionally on the latter, an angle $\boldsymbol{\theta}$ is sampled from a multivariate normal distribution. The product of the two gives the desired multivariate regularly-varying vector. The appropriate KL-divergences are derived leading to two objective functions: one for the radius VAE and the other one for the angular VAE.

6.2.2 d-max-decreasing neural networks

In [36], the authors propose a new neural network architecture called d-max-decreasing neural network and inspired by Maxout networks [34]. This architecture naturally encodes the constraints associated with an angular measure (26) and therefore the outputs of the neural network are simulated from a valid multivariate extreme-value distribution. The proof of an approximation rate is part of the authors' future work.

Table 4: Estimated Kendall's tau computed with the GAN approach on simulated bivariate data.

Tail-index	2nd-order parameters	Theoretical Kendall's tau		
γ	(ρ_1, ρ_2)	$\tau_{C^G_{1,1}} = 0.1$	$\tau_{C^G_2} = 0.5$	$\tau_{C^G_{10}} = 0.9$
0.1	$(-1, -2)$	0.092	0.514	0.905
	$(-1, -3)$	0.093	0.477	0.900
	$(-2, -3)$	0.086	0.511	0.899
0.5	$(-1, -2)$	0.090	0.493	0.903
	$(-1, -3)$	0.106	0.506	0.901
	$(-2, -3)$	0.093	0.473	0.885
0.9	$(-1, -2)$	0.088	0.500	0.903
	$(-1, -3)$	0.091	0.484	0.899
	$(-2, -3)$	0.073	0.487	0.900

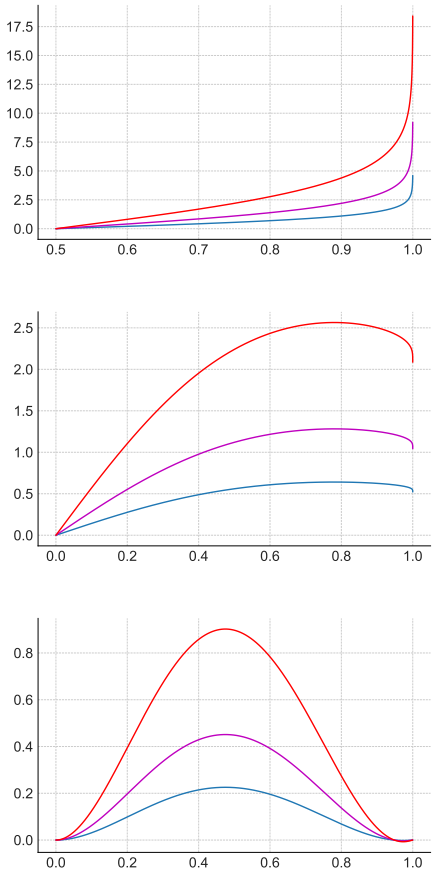

Figure 6: Log-quantile function $u \in (1/2, 1) \mapsto \log q_X(u)$ (top panel), tail-index function $u \in (0, 1) \mapsto f^{\mathrm{TIF}}(u)$ (middle panel) and corrected tail-index function $u \in (0, 1) \mapsto f^{\mathrm{CTIF}}(u)$ (bottom panel) associated with the Burr$(\gamma, \rho = -1)$ distribution with tail-index $\gamma = 1/2$ (blue), $\gamma = 1$ (purple) and $\gamma = 2$ (red).

7 Simulating extremes with GANs, numerical illustrations in higher dimension

The ability of GAN and of its refinement EV-GAN to properly scale in high dimension is now investigated. Using the `R` package `copulas` [45], $n = 10,000$ samples are simulated from the d-variate Gumbel copula $C^{\mathrm{G}}_{\mu=2}$ for increasing dimensions $d \in \{4, 8, 16, 32, 64, 128, 256, 512\}$ and with Burr$(\gamma = 1/2, \rho = -1)$ margins. MSLE(ξ) at level $\xi \in \{0.90, 0.95, 0.99\}$ are reported in Table 5 for both GAN and EV-GAN methods. EV-GAN yields realistic margins for all considered dimensions and for high levels of quantiles $\xi \in \{0.90, 0.95\}$. In the case of higher levels ($\xi \in \{0.99\}$) the dimension is limited to 128. In contrast, the classic GAN model is limited to a dimension about 8 for all considered levels ξ.

Figure 7 illustrates the dependence associated with samples in dimension $d \in \{4, 8, 16, 32, 64, 128\}$. First, remark that the $\lambda(\cdot)$ function associated with the original data tends toward the asymptotic independence function $\lambda_{\Pi,\infty}(\cdot)$ as d increases, accordingly to [28, Section 3.3]. Second, it appears that EV-GAN manages to reproduce well the dependence structure of the original data up to $d = 16$, but tends to the independence between the margins for higher dimensions.

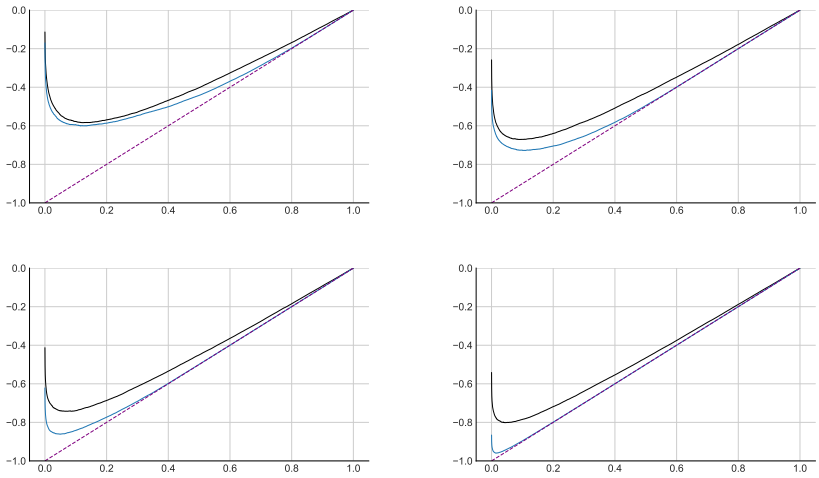

Figure 7: Estimated $t \in [0, 1] \mapsto \lambda(t)$ functions on d-variate simulated data with $d \in \{16, 32, 64, 128\}$ (from left to right and top to bottom). Black: original sample, blue: sample from the EV-GAN model, dashed purple: asymptotic independence case $t \mapsto \lambda_{\Pi,\infty}(t) = t - 1$.

Table 5: Performance comparison between GAN and EV-GAN on simulated d-variate data with respect to the MSLE(ξ) criteria computed at levels $\xi \in \{0.90, 0.95, 0.99\}$, MSLE($\xi$) ≥ 1 are not reported.

dimension	MSLE(0.90)		MSLE(0.95)		MSLE(0.99)	
d	GAN	EV-GAN	GAN	EV-GAN	GAN	EV-GAN
4	0.065	0.020	0.117	0.036	0.353	0.125
8	0.237	0.071	0.366	0.109	-	0.264
16	0.991	0.235	-	0.264	-	0.198
32	0.988	0.261	0.908	0.209	-	0.666
64	-	0.280	-	0.265	-	0.318
128	-	0.403	-	0.307	-	0.603
256	-	0.376	-	0.642	-	-
512	-	0.404	-	0.393	-	-

8 Conclusion, discussion and further reading

Neural network generative modeling is increasingly attracting attention based on impressive empirical results. Focusing on (2), this problem first requires information on the quantity of interest (regularity, structure) in order to give theoretical guidelines to build, in a second time, an appropriate generator with desired properties (convergence and stability of the optimization process, richness of the generated samples). When the target random quantity is supposed to be heavy-tailed, we have shown (theoretically and on several data experiments) that usual GAN models cannot reproduce this property without dedicated architecture improvements. An overview of a number of proposals along these lines has been proposed in this article for GAN and VAE frameworks.

Diffusion models [37, 59] are the most recent class of generative neural networks. The dynamics of the diffusion (forward and backward) are parameterized by neural networks in the drift of a Gaussian noise (see Paragraph 2.5). It would be interesting to investigate whether these new generative methods are able to simulate realistic tail events. This is of primary importance in risk assessment where simulating too light-tailed events may yield a severe underestimation of extreme risks.

We also believe that neural networks can reveal useful to estimate tail quantities such as extreme risk measures (see Paragraph 5.3). One can for instance increase the sample size using generative methods so that the estimation does not require extrapolation any more. Another solution explored in [5, 6] is to exploit the powerful optimization techniques associated with neural networks to fit higher order extreme-value models in order to reduce the estimation bias.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. `https://www.tensorflow.org/`.

[2] C. Acerbi. Spectral measures of risk: A coherent representation of subjective risk aversion. *Journal of Banking and Finance*, 26(7):1505–1518, 2002.

[3] C. Acerbi and B. Szekely. Back-testing expected shortfall. *Risk*, 27(11):76–81, 2014.

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

[5] M. Allouche, S. Girard, and E. Gobet. Learning extreme expected shortfall with neural networks. `https://hal.archives-ouvertes.fr/hal-03751980`, 2023.

[6] M. Allouche, S. Girard, and E. Gobet. Estimation of extreme quantiles from heavy-tailed distributions with neural networks. *Statistics and Computing*, 34:12, 2024.

[7] B.D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[8] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th Int. Conf. on Mach. Learn.*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 2017.

[9] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999.

[10] C. Bernard and C. Czado. Conditional quantiles and tail dependence. *Journal of Multivariate Analysis*, 138:104–126, 2015.

[11] D. P. Bertsekas and S. E. Shreve. *Stochastic optimal control*, volume 139 of *Mathematics in Science and Engineering*. Academic Press, Inc., New York-London, 1978.

[12] S. Bhatia, A. Jain, and B. Hooi. ExGAN: Adversarial generation of extreme samples. arXiv preprint `arXiv:2009.08454`, 2020.

[13] G. Biau, B. Cadre, M. Sangnier, and U. Tanielian. Some theoretical properties of GANs. *The Annals of Statistics*, 48(3):1539–1566, 2020.

[14] G. Biau, M. Sangnier, and U. Tanielian. Some theoretical insights into Wasserstein GANs. arXiv preprint `arXiv:2006.02682`, 2020.

[15] N. H. Bingham, C. M. Goldie, and J. L. Teugels. *Regular variation*, volume 27 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1987.

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

[17] R. Cont, M. Cucuringu, R. Xu, and C. Zhang. Tail-GAN: Nonparametric scenario generation for tail risk estimation. arXiv preprint `arXiv:2203.01664`, 2022.

[18] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control Signals Systems*, 2(4):303–314, 1989.

[19] V. De Bortoli, J. Thornton, J. Heng, and A. Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

[20] L. de Haan and A. Ferreira. *Extreme value theory*. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2006.

[21] P. K. Diederik and J. Ba. Adam: A method for stochastic optimization. arXiv preprint `arXiv:1412.6980`, 2017.

[22] C. Doersch. Tutorial on variational autoencoders. arXiv preprint `arXiv:1606.05908`, 2016.

[23] R. Eckhardt. Stam Ulam, John Von Neumann and the Monte-Carlo method. *Los Alamos Science*, Special Issue:131–143, 1987.

[24] J. H. J. Einmahl and J. Segers. Maximum empirical likelihood estimation of the spectral measure of an extreme-value distribution. *The Annals of Statistics*, 37(58):2953–2989, 2009.

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

[26] D. Foster. *Generative deep learning: teaching machines to paint, write, compose, and play*. O'Reilly Media, 2019.

[27] A R. Gallant and H. White. There exists a neural network that does not make avoidable mistakes. In *IEEE 1988 International Conference on Neural Networks*, pages 657–664, 1988.

[28] M. Garcin, D. Guegan, and B. Hassani. A novel multivariate risk measure: the Kendall VaR. `https://halshs.archives-ouvertes.fr/halshs-01467857`, 2018.

[29] C. Genest and J. MacKay. The joy of copulas: bivariate distributions with uniform marginals. *The American Statistician*, 40(4):280–283, 1986.

[30] C. Genest and L.-P. Rivest. A characterization of Gumbel's family of extreme value distributions. *Statistics and Probability Letters*, 8(3):207–211, 1989.

[31] C. Genest and L.-P. Rivest. Statistical inference procedures for bivariate Archimedean copulas. *Journal of American Statistical Association*, 88(423):1034–1043, 1993.

[32] G. M. Goerg. The Lambert way to gaussianize heavy-tailed data with the inverse of Tukey's h transformation as a special case. *The Scientific World Journal*, ID 909231, 2015.

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680, 2014.

[34] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1319–1327. PMLR, 2013.

[35] M. Haas and S. Richter. Statistical analysis of Wasserstein GANs with applications to time series forecasting. arXiv preprint `arXiv:2011.03074`, 2020.

[36] A. Hasan, K. Elkhalil, Y. Ng, J. M. Pereira, S. Farsiu, J. H. Blanchet, and V. Tarokh. Modeling extremes with d-max-decreasing neural networks. arXiv preprint `arXiv:2102.09042`, 2022.

[37] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[38] M. Hofert. Sampling Archimedean copulas. *Computional Statistics and Data Analysis*, 52(12):5163–5174, 2008.

[39] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[40] T. Huster, J. E. J. Cohen, Z. Lin, K. Chan, C. Kamhoua, N. Leslie, C. Y. J. Chiang, and V. Sekar. Pareto GAN: Extending the representational power of GANs to heavy-tailed distributions. arXiv preprint `arXiv:2101.09113`, 2021.

[41] M. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[42] D. P. Kingma and M. Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[43] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. arXiv preprint `arXiv:1312.6114`, 2022.

[44] E. Kohlbecker. Weak asymptotic properties of partitions. *Transactions of The American Mathematical Society*, 88(2):346–365, 1958.

[45] I. Kojadinovic and J. Yan. Modeling multivariate distributions with continuous margins using the copula R package. *Journal of Statistical Software*, 34(9):1–20, 2010.

[46] W. Kritzinger, M. Karner, J. Traar, G.and Henjes, and W. Sihn. Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51(11):1016–1022, 2018.

[47] N. Lafon, P. Naveau, and R. Fablet. A VAE approach to sample multivariate extremes. arXiv preprint `arXiv:2306.10987`, 2023.

[48] M. Leshno, W.Y. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

[49] A. McNeil and J. Nešlehová. Multivariate Archimedean copulas, d-monotone functions and l_1-norm symmetric distributions. *The Annals of Statistics*, 37(5B):3059–3097, 2009.

[50] J. El Methni, L. Gardes, and S. Girard. Nonparametric estimation of extreme risks from conditional heavy-tailed distributions. *Scandinavian Journal of Statistics*, 41:988–1012, 2014.

[51] E. Negri, L. Fumagalli, and M. Macchi. A review of the roles of digital twin in CPS-based production systems. *Procedia Manufacturing*, 11:939–948, 2017.

[52] R. Nelsen. *An introduction to copulas*. Springer Series in Statistics. Springer, New York, second edition, 2006.

[53] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner,

L. Fang, J. Bai, and Soumith C. 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, *33rd Conference on Neural Information Processing Systems*, pages 8024–8035. Curran Associates, Inc., 2019.

[54] A. Pinkus. Approximation theory of the MLP model in neural networks. In *Acta numerica*, volume 8, pages 143–195. Cambridge University Press, Cambridge, 1999.

[55] A. Pinkus. *Ridge functions*, volume 205 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2015.

[56] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1530–1538. PMLR, 2015.

[57] R. T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, 26(7):1443–1471, 2002.

[58] A. Sklar. Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris*, 8:229–231, 1959.

[59] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 2256–2265. PMLR, 2015.

[60] C. Villani. *Optimal transport*, volume 338 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 2009.

[61] M. Vladimirova, S. Girard, N. Hien, and J. Arbel. Sub-Weibull distributions: generalizing sub-Gaussian and sub-Exponential properties to heavier-tailed distributions. *Stat*, 9:e318, 2020.

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

[63] F. Wu, E. Valdez, and M. Sherris. Simulating from exchangeable Archimedean copulas. *Communications in Statistics - Simulation and Computation*, 36(5):1019–1034, 2007.