Simulation of extreme values with Generative Adversarial Networks

Michaël Allouche Stéphane Girard Emmanuel Gobet

Workshop of the Chaire Assurabilité des Risques Emergents: Generative AI for Extreme Events
Sorbonne University, Paris, France

March 19, 2026

Motivations

▷ **Risk assessment**: extreme values have low probability / high impact.

credit: edelit

credit: latimes

▷ Estimation of **extreme risk measures** (such as $q(\alpha)$ the quantile / VaR / return level) requires dedicated methods: when $\alpha \to 1$,

$$\mathbb{P}(q(\alpha) > \max_{i=1,\dots,n} X_i) = \mathbb{P}(q(\alpha) > X_1)^n = \alpha^n = e^{-n(1-\alpha)(1+o(1))} \to 1$$

if $n(1-\alpha) \to 0$. The quantile is almost surely asymptotically outside the observation range $\{X_1, \dots, X_n\}$.

Two lines of work

– Rare event type simulation methods –

▷ Existing simulation methods (Importance Sampling, MCMC) work well but high complexity cost.

▷ Focus on new **data-based generative models** trained on

- true data sets (*i.e.* historical financial returns),
- outputs of black-box models (*i.e.* meteorological models).

▷ EV-GAN: [Allouche et al., 2022]

▷ ExceedGAN:[Allouche et al., 2026]

– Extrapolation based on Extreme-Value Theory (EVT) –

▷ **Higher order** extrapolation methods leveraging the approximation properties of **neural networks** [Allouche et al., 2024].

▷ Extension to estimate **extreme Expected Shortfall** (and even more generally **extreme conditional tail moments**) as functions of confidence levels [Allouche et al., 2025].

Ingredients: Generative Modeling, Neural Networks, Extreme Value Theory

Generative modeling

If $\mathbf{X}$ denotes the r.v. taking values in some space $\mathcal{X} \subseteq \mathbb{R}^d$ from which we have observations $(\mathbf{X}_1, \ldots, \mathbf{X}_n)$, the problem is to find a function $G : \mathcal{Z} \to \mathcal{X}$ and a **latent probability distribution** p_Z on $\mathcal{Z} \subseteq \mathbb{R}^{d_Z}$ s.t.

$$\mathbf{X} \overset{\mathrm{d}}{=} G(\mathbf{Z}) \text{ and } \mathbf{Z} \sim p_Z \tag{1}$$

Generative modeling

If $\mathbf{X}$ denotes the r.v. taking values in some space $\mathcal{X} \subseteq \mathbb{R}^d$ from which we have observations $(\mathbf{X}_1, \dots, \mathbf{X}_n)$, the problem is to find a function $G : \mathcal{Z} \to \mathcal{X}$ and a **latent probability distribution** p_Z on $\mathcal{Z} \subseteq \mathbb{R}^{d_Z}$ s.t.

$$\mathbf{X} \overset{\mathrm{d}}{=} G(\mathbf{Z}) \text{ and } \mathbf{Z} \sim p_Z \tag{1}$$

▷ **Which class of functions G and densities p_Z may be considered to ensure that (1) holds?**

Generative modeling

If $\mathbf{X}$ denotes the r.v. taking values in some space $\mathcal{X} \subseteq \mathbb{R}^d$ from which we have observations $(\mathbf{X}_1, \ldots, \mathbf{X}_n)$, the problem is to find a function $G : \mathcal{Z} \to \mathcal{X}$ and a **latent probability distribution** p_Z on $\mathcal{Z} \subseteq \mathbb{R}^{d_Z}$ s.t.

$$\mathbf{X} \stackrel{\mathrm{d}}{=} G(\mathbf{Z}) \text{ and } \mathbf{Z} \sim p_Z \tag{1}$$

▷ **Which class of functions G and densities p_Z may be considered to ensure that (1) holds?**

Theorem (Kuratowski [Villani, 2009])

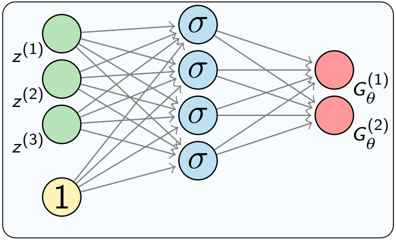
Let $(\mathcal{Z}, \mu_Z)$ and $(\mathcal{X}, \mu_X)$ two Polish probability spaces. Then there exists a (non-unique) measurable bijection G s.t. (1) holds.

Neural Networks

▷ **How to build an approximation of G ?**

Neural Networks

▷ How to build an approximation of G ?



Generator with $d_Z = 3, J = 4$ *and* $d = 2$

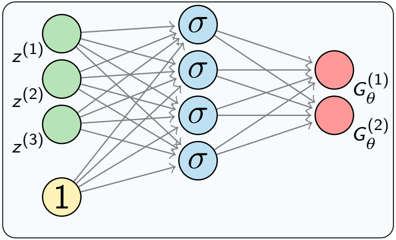
$$G_\theta^{(m)}\left(z^{(1)}, \dots, z^{(d_Z)}\right) = \sum_{j=1}^{\mathbf{J}} a_j^{(m)} \sigma\left(\sum_{i=1}^{d_Z} w_j^{(i)} z^{(i)} + b_j\right)$$

with $m \in \{1, \dots, d\}$

and $\sigma(\cdot)$ a **non-linear function**, e.g. $\sigma(x) = x_+$

Neural Networks

▷ How to build an approximation of G ?



Generator with $d_Z = 3, J = 4$ *and* $d = 2$

$$G_\theta^{(m)}\left(z^{(1)},\dots,z^{(d_Z)}\right) = \sum_{j=1}^{J} a_j^{(m)} \sigma\left(\sum_{i=1}^{d_Z} w_j^{(i)} z^{(i)} + b_j\right)$$

with $m \in \{1,\dots,d\}$

and $\sigma(\cdot)$ a **non-linear function**, e.g. $\sigma(x) = x_+$

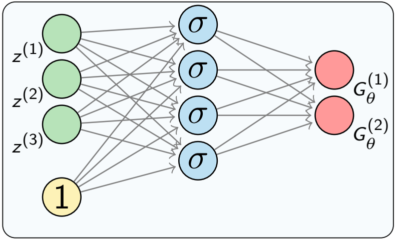
Theorem (Universal Approximation Theorem [Pinkus, 1999])

Suppose G is a **continuous** *function on a* **compact** *space $\mathcal{Z} \subset \mathbb{R}^{d_Z}$ and σ is not a polynomial, then $\forall \varepsilon > 0$, there exists a NN G_θ s.t.*

$$\sup_{\mathbf{z}\in\mathcal{Z}} |G(\mathbf{z}) - G_\theta(\mathbf{z})| < \varepsilon. \tag{2}$$

Neural Networks

▷ How to build an approximation of G ?



Generator with $d_Z = 3, J = 4$ *and* $d = 2$

$$G_\theta^{(m)}\left(z^{(1)}, \dots, z^{(d_Z)}\right) = \sum_{j=1}^{J} a_j^{(m)} \sigma \left(\sum_{i=1}^{d_Z} w_j^{(i)} z^{(i)} + b_j \right)$$

with $m \in \{1, \dots, d\}$

and $\sigma(\cdot)$ a **non-linear function**, e.g. $\sigma(x) = x_+$

Theorem (Universal Approximation Theorem [Pinkus, 1999])

Suppose G *is a* **continuous** *function on a* **compact** *space* $\mathcal{Z} \subset \mathbb{R}^{d_Z}$ *and* σ *is not a polynomial, then* $\forall \varepsilon > 0$, *there exists a NN* G_θ *s.t.*

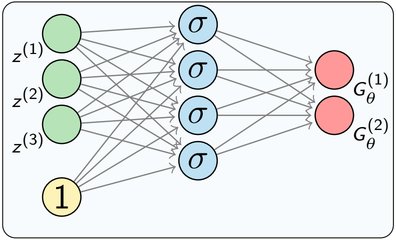
$$\sup_{\mathbf{z} \in \mathcal{Z}} |G(\mathbf{z}) - G_\theta(\mathbf{z})| < \varepsilon. \tag{2}$$

Theorem (Rate of convergence [Maiorov, 1999])

Given a regularity r *and a dimension* d, (2) *holds with a rate* $\sim J^{-r/d}$.

Neural Networks

▷ How to build an approximation of G ?



Generator with $d_Z = 3$, $J = 4$ and $d = 2$

$$G_\theta^{(m)}\left(z^{(1)}, \ldots, z^{(d_Z)}\right) = \sum_{j=1}^{J} a_j^{(m)} \sigma \left(\sum_{i=1}^{d_Z} w_j^{(i)} z^{(i)} + b_j \right)$$

with $m \in \{1, \ldots, d\}$
and $\sigma(\cdot)$ a **non-linear function**, e.g. $\sigma(x) = x_+$

▷ How to learn G_θ ?

Theorem (Universal Approximation Theorem [Pinkus, 1999])

Suppose G is a **continuous** *function on a* **compact** *space $\mathcal{Z} \subset \mathbb{R}^{d_Z}$ and σ is not a polynomial, then $\forall \varepsilon > 0$, there exists a NN G_θ s.t.*

$$\sup_{\mathbf{z} \in \mathcal{Z}} |G(\mathbf{z}) - G_\theta(\mathbf{z})| < \varepsilon. \quad (2)$$

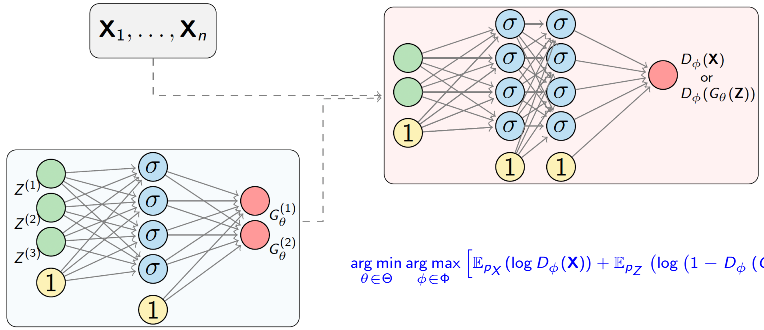
Theorem (Rate of convergence [Maiorov, 1999])

Given a regularity r and a dimension d, (2) holds with a rate $\sim J^{-r/d}$.

Generative Adversarial Networks (GANs)

GANs. [Goodfellow et al., 2014]

- **Generator**: Learn $\theta^\star$ s.t. $G_{\theta^\star}(\mathbf{Z}) \overset{\mathrm{d}}{\approx} \mathbf{X}$
- **Discriminator**: Learn $\phi^\star$ s.t. $D_{\phi^\star}(\mathbf{X}) \simeq 1$ and $D_{\phi^\star}(G_\theta(\mathbf{Z})) \simeq 0$

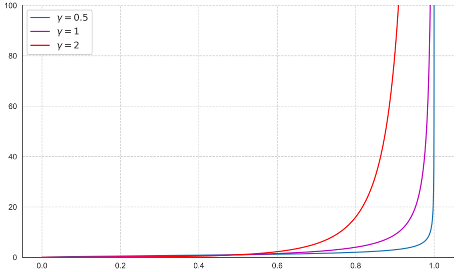


$$\arg\min_{\theta\in\Theta} \arg\max_{\phi\in\Phi} \left[\mathbb{E}_{p_X}(\log D_\phi(\mathbf{X})) + \mathbb{E}_{p_Z}\left(\log\left(1 - D_\phi\left(G_\theta(\mathbf{Z})\right)\right)\right)\right]$$

Problem statement

($\mathbf{H_1}$): Focusing on heavy-tailed distributions ($F \in \mathrm{MDA}(\text{Fréchet})$), the tail quantile function $U(t) := q(1-1/t), \forall t > 1$, is **regularly varying** with tail index $\gamma > 0$ ($U \in \mathcal{RV}_\gamma$) and $U(t) = t^\gamma L(t)$ with $L \in \mathcal{RV}_0$, *i.e.*

$$L(\lambda t)/L(t) \to 1 \text{ as } t \to \infty, \forall \lambda > 0.$$

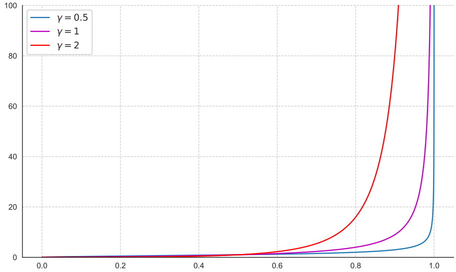


Quantile function of a Burr distribution $u \mapsto q(u)$ with parameters $\gamma = \{0.5, 1, 2\}$ and $\rho = -1$

Problem statement

$(\mathbf{H_1})$: Focusing on heavy-tailed distributions ($F \in \mathrm{MDA}(\text{Fréchet})$), the tail quantile function $U(t) := q(1-1/t), \forall t > 1$, is **regularly varying** with tail index $\gamma > 0$ ($U \in \mathcal{RV}_\gamma$) and $U(t) = t^\gamma L(t)$ with $L \in \mathcal{RV}_0$, *i.e.*

$$L(\lambda t)/L(t) \to 1 \text{ as } t \to \infty, \forall \lambda > 0.$$



Quantile function of a Burr distribution $u \mapsto q(u)$ **with parameters** $\gamma = \{0.5, 1, 2\}$ **and** $\rho = -1$

⚠ **Challenges** ⚠

- The UAT doesn't guarentee good guarentee accuracy in the tail
- If Z is either bounded or a Gaussian vector, by no means $G_\theta(\mathbf{Z}) \stackrel{\mathrm{d}}{=} X$

GANs: Solutions proposed in the literature

- Use a light-tail model for Z on transformed data and then transform back the generator outputs for recovering the heavy-tail property:
  – Lambert fct $W(x) = x \exp(\gamma x^2/2)$ → Quant-GAN [Wiese et al., 2020],
  – cdf of the $\text{EVD}(\gamma)$ → evtGAN [Boulaguiem et al., 2022].
  In both cases, the tail-index γ has to be estimated first.
- Use a heavy-tailed distribution for Z:
  – Student-t [Feder et al., 2020],
  – $\text{GPD}(\gamma)$ → Pareto-GAN [Huster et al., 2021].
  In both cases, the tail-index γ has to be estimated first.
- EVT on margins, GAN on dependence → WA-GAN [Lhaut et al., 2026].
- New cost function: Exploit the joint elicitability property of the VaR and ES risk measures → Tail-GAN [Cont et al., 2022].
- Heuristics: → Ex-GAN [Bhatia et al., 2020].

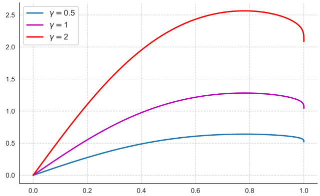
Other architectures.

- Variational Auto Encoders (VAE) [Lafon et al., 2023],
- d-max-decreasing neural networks [Hasan et al., 2022].

EV-GAN: Tail-index function (TIF)

Since $q(1-1/\cdot) \in \mathcal{RV}_\gamma$, then for all $u \in (0,1)$, $q(u) = (1-u)^{-\gamma} L\left(\frac{1}{1-u}\right)$, with $L \in \mathcal{RV}_0$.

$$f^{\mathrm{TIF}}(u) = -\frac{\log\left(q(u)\right)}{\log\left(\frac{1-u^2}{2}\right)}$$



TIF associated with Burr distribution $(\rho = -1)$

▷ Taking the log yields $\log q(u) \sim -\gamma \log(1-u)$ as $u \to 1$, since $\log L(v)/\log v \to 0$ as $v \to \infty$ [Bingham et al., 1987, Proposition 1.3.6];

▷ Note also that $\log((1-u^2)/2) \sim \log(1-u)$ as $u \to 1$;

▷ As a conclusion: $f^{\mathrm{TIF}}(u) \to \gamma$ as $u \to 1$ and $f^{\mathrm{TIF}}(0) = 0$.

EV-GAN: Second-order condition

Second order condition. There exist $\rho < 0$ and $|A| \in \mathcal{RV}_\rho$ s.t.

$$\frac{1}{A(t)} \log\left(\frac{L(zt)}{L(t)}\right) \longrightarrow \frac{z^\rho - 1}{\rho}, \quad \text{as } t \to \infty, \ \forall z \geq 1.$$

The second-order parameter ρ tunes the rate of convergence of $L(zt)/L(t)$ to 1.

EV-GAN: Second-order condition

Second order condition. There exist $\rho < 0$ and $|A| \in \mathcal{RV}_\rho$ s.t.

$$\frac{1}{A(t)} \log\left(\frac{L(zt)}{L(t)}\right) \longrightarrow \frac{z^\rho - 1}{\rho}, \quad \text{as } t \to \infty,\ \forall z \geq 1.$$

The second-order parameter ρ tunes the rate of convergence of $L(zt)/L(t)$ to 1.

Karamata representation([de Haan and Ferreira, 2006, Definition B 1.6])**.**
The function $L \in \mathcal{RV}_0$ if and only if L can be written as

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

where $c(x) \to c_\infty$ and $\varepsilon(x) \to 0$ as $x \to \infty$.

EV-GAN: Second-order condition

Second order condition. There exist $\rho < 0$ and $|A| \in \mathcal{RV}_\rho$ s.t.

$$\frac{1}{A(t)} \log\left(\frac{L(zt)}{L(t)}\right) \longrightarrow \frac{z^\rho - 1}{\rho}, \quad \text{as } t \to \infty,\ \forall z \geq 1.$$

The second-order parameter ρ tunes the rate of convergence of $L(zt)/L(t)$ to 1.

Karamata representation([de Haan and Ferreira, 2006, Definition B 1.6]).
The function $L \in \mathcal{RV}_0$ if and only if L can be written as

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

where $c(x) \to c_\infty$ and $\varepsilon(x) \to 0$ as $x \to \infty$.

$(\mathbf{H_2})$: $c(x) = c_\infty$ for all $x \geq 1$ and $\varepsilon(x) = x^\rho \ell(x)$ with $\ell \in \mathcal{RV}_0$ and $\rho < 0$ is referred to the **second-order parameter**

EV-GAN: Second-order condition

Second order condition. There exist $\rho < 0$ and $|A| \in \mathcal{RV}_\rho$ s.t.

$$\frac{1}{A(t)} \log\left(\frac{L(zt)}{L(t)}\right) \longrightarrow \frac{z^\rho - 1}{\rho}, \quad \text{as } t \to \infty, \ \forall z \geq 1.$$

The second-order parameter ρ tunes the rate of convergence of $L(zt)/L(t)$ to 1.

Karamata representation([de Haan and Ferreira, 2006, Definition B 1.6]).
The function $L \in \mathcal{RV}_0$ if and only if L can be written as

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

where $c(x) \to c_\infty$ and $\varepsilon(x) \to 0$ as $x \to \infty$.

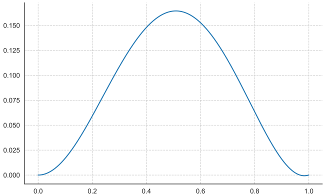
$(\mathbf{H_2})$: $c(x) = c_\infty$ for all $x \geq 1$ and $\varepsilon(x) = x^\rho \ell(x)$ with $\ell \in \mathcal{RV}_0$ and $\rho < 0$ is referred to the **second-order parameter**

$$\Rightarrow \partial_u f^{\mathrm{TIF}}(u) = \sum_{j=0}^{3} c_j \varphi_j(u) - \frac{\varepsilon\left(\frac{1}{(1-u)}\right)}{(1-u)\log(1-u)}(1 + o(1)) + \ldots, \text{as } u \to 1$$

EV-GAN: Corrected TIF (CTIF)

Introduce for all $u \in (0,1)$

$$f^{\mathrm{CTIF}}(u) = f^{\mathrm{TIF}}(u) - \sum_{k=1}^{6} \kappa_k e_k(u)$$



CTIF with $\gamma = 0.5$ *and* $\rho = -3$

where

- $\{\kappa_1, \dots, \kappa_6\}$ are coefficients depending on the distribution F,
- $\{e_1, \dots, e_6\}$ are fixed functions independent from F.

This correction allows us to get higher regularity provided ρ is not too large.

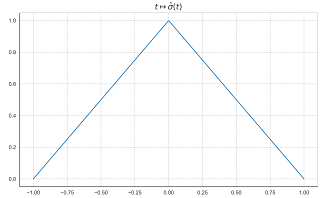
Proposition

(i) If $(\mathbf{H_1})$ *holds, then* $f^{\mathrm{CTIF}} \in \mathcal{C}^0([0,1])$.
(ii) If, moreover, $(\mathbf{H_2})$ *holds with* $\rho < -1$*, then* $f^{\mathrm{CTIF}} \in \mathcal{C}^1([0,1])$.
(iii) If, moreover, $\rho < -2$*, then* $f^{\mathrm{CTIF}} \in \mathcal{C}^2([0,1])$.

EV-GAN: Approximation error with ReLUs

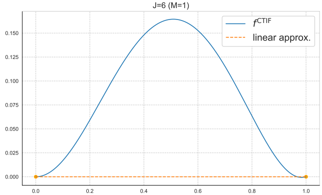
Triangular function. Built with 3 translated ReLU $\sigma(x) := \max(0, x)$

$$\hat{\sigma}(t) := \sigma^{\mathrm{R}}(t+1) - 2\sigma^{\mathrm{R}}(t) + \sigma^{\mathrm{R}}(t-1)$$



Piecewise linear approximation. For all $M \in \mathbb{N}_0$, let $\delta = 1/M$, $t_j = j/M$ for $j = 0, \dots, M$. If $f \in \mathcal{C}^2([0,1])$, then

$$\sup_{t\in[0,1]} \left| f(t) - \sum_{j=0}^{M} f(t_j)\hat{\sigma}\left(\frac{t-t_j}{\delta}\right) \right| \leq cM^{-2}$$



- Approx. order M^{-2} with $J = 3(M+1)$ ReLUs.
- Similar result for $\mathcal{C}^{1,\alpha}$ functions, $\alpha \in (0,1)$.

$\gamma = 0.5$, $\rho = -3$

EV-GAN: Main result

Applying the previous construction to the CTIF yields:

Theorem

Assume $(\mathbf{H_1})$ *and* $(\mathbf{H_2})$ *hold. For all* $m \in \{1, \ldots, d\}$ *with* $\rho^{(m)} < -2$*, consider the generator:*

$$G_\theta^{(m)}(z) = \sum_{j=1}^{J} a_j^{(m)} \sigma^{\mathrm{R}} \left(\sum_{i=1}^{d_z} w_j^{(i)} z^{(i)} + b_j \right) + \sum_{k=1}^{6} \kappa_k^{(m)} e_k \left(z^{(m)} \right),$$

where $\theta = \left(a_j^{(m)}, b_j, w_j^{(i)}, \kappa_k^{(m)} \right)_{i,j,k,m}$. *If* $\rho^{(m)} < -2$ *then*

$$\inf_\theta \max_{m \in \{1,\ldots,d\}} \sup_{z \in [0,1]^{d'}} \left| f^{\mathrm{TIF},(m)}(z^{(m)}) - G_\theta^{(m)}(z) \right| = \mathcal{O}\left(J^{-2} \right).$$

Remark. Similar result when $-2 \leq \rho < -1$ with reduced rate of convergence.

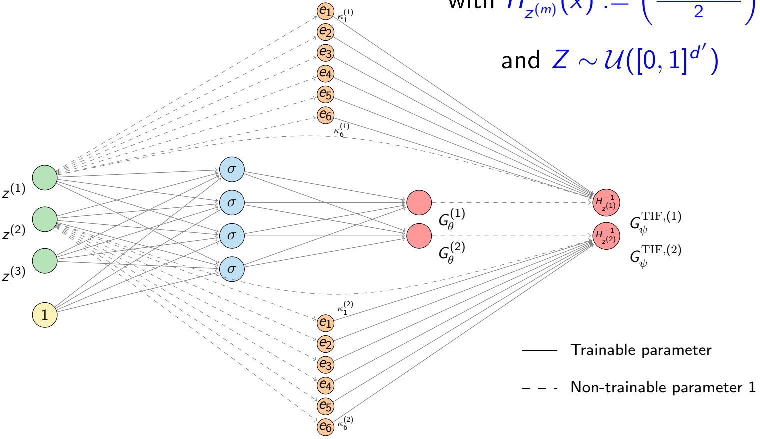
EV-GAN: Architecture

In a multidimensional setting $d > 1$ and $d' > 1$, for each m-th marginal:

$$G_{\psi}^{\mathrm{TIF},(m)}(z) = H_{z^{(m)}}^{-1}\left(G_{\theta}^{(m)}(z) + \sum_{k=1}^{6} \kappa_k^{(m)} e_k\left(z^{(m)}\right)\right), \quad (3)$$

with $H_{z^{(m)}}^{-1}(x) := \left(\frac{1-(z^{(m)})^2}{2}\right)^{-x}$

and $Z \sim \mathcal{U}([0,1]^{d'})$



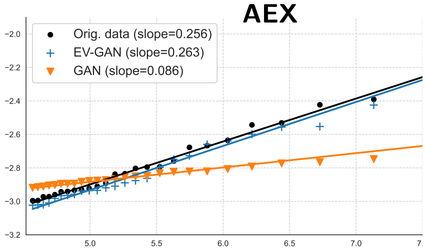
Generator of the EV-GAN with $d_z = 3$ and $d = 2$

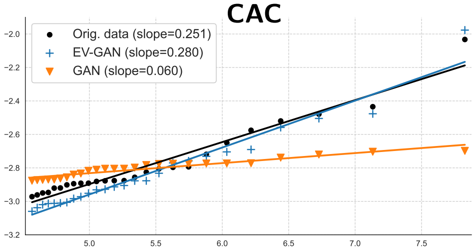
EV-GAN: Illustration on European stock indices

Setting.

- Log-returns on selected heavy-tailed financial stock market indices.
- Processing of missing data: compute and synchronize log-returns.

Margins fit: log quantile-quantile plot $-\log(1-u) \mapsto \log q(u)$ *for* $u \geq 0.99$.





EV-GAN: Illustration on European stock indices

Performance assessment.

- Marginals: Mean Squared Logarithmic Error

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

- Dependence: Absolute Kendall Error (AKE).

ticker	NKX		Europe		Asia		World	
dimension d	1		2		3		6	
sample size n	3173		2504		1378		548	
MSLE(0.90)	0.010	**0.003**	0.080	**0.003**	0.049	**0.014**	0.069	**0.018**
MSLE(0.95)	0.015	**0.002**	0.103	**0.004**	0.031	**0.012**	0.093	**0.025**
MSLE(0.99)	0.029	**0.004**	0.058	**0.007**	0.022	**0.011**	0.104	**0.041**
AKE	–	–	16.807	**4.697**	9.760	**4.872**	24.781	**3.533**

Performance comparison between the best GAN (left) and EV-GAN (right). The best result is emphasized in bold

ExceedGAN: Motivations

Build a GAN to sample new data points (called **exceedances**) from

$$\mathbf{X} \mid \mathbf{X} > \mathbf{u}$$

(componentwisely) where $\mathbf{u}$ is a **high multivariate** threshold.

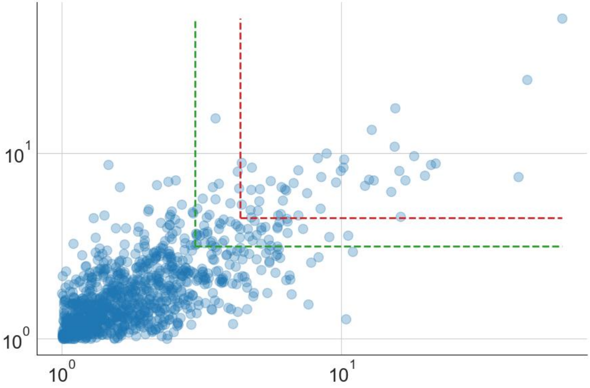


Illustration for two bivariate thresholds u.

ExceedGAN: Statistical framework $(d = 1)$

Given an independent sample $\{X_1, \ldots, X_n\}$ from F_X, we focus on the simulation of $Y(\delta_n) = X \mid X > q_X(1 - \delta_n)$, where $\delta_n \to 0$ as $n \to \infty$.

ExceedGAN: Statistical framework ($d = 1$)

Given an independent sample $\{X_1, \ldots, X_n\}$ from F_X, we focus on the simulation of $Y(\delta_n) = X \mid X > q_X(1 - \delta_n)$, where $\delta_n \to 0$ as $n \to \infty$.

Lemma

Assume that the distribution of X is continuous. Then, for all $\delta_n \in (0, 1)$

$$Y(\delta_n) \stackrel{\mathrm{d}}{=} q_{Y(\delta_n)}(1 - Z) = q_X(1 - \delta_n Z),$$

with $Z \sim \mathcal{U}([0, 1])$.

ExceedGAN: Statistical framework ($d = 1$)

Given an independent sample $\{X_1, \ldots, X_n\}$ from F_X, we focus on the simulation of $Y(\delta_n) = X \mid X > q_X(1-\delta_n)$, where $\delta_n \to 0$ as $n \to \infty$.

Lemma

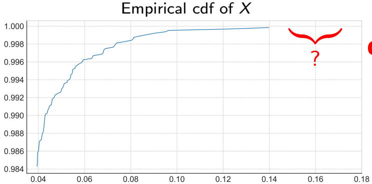
Assume that the distribution of X is continuous. Then, for all $\delta_n \in (0,1)$

$$Y(\delta_n) \stackrel{\mathrm{d}}{=} q_{Y(\delta_n)}(1-Z) = q_X(1-\delta_n Z),$$

with $Z \sim \mathcal{U}([0,1])$.

⚠ Challenge ⚠

For small values of z or δ_n, $q_X(1-\delta_n z)$ is an **extreme quantile** likely to be **larger sample maximum**



out-of-sample

ExceedGAN: Extrapolation principle

Take advantage of $U_X(t) = t^\gamma L(t)$ ($U_X \in \mathcal{RV}_\gamma$) to link the extreme quantile

$$q_{Y(\delta_n)}(1 - z) = q_X(1 - \delta_n z) = U_X(1/(\delta_n z)),$$

and the threshold $q_X(1 - \delta_n) = U_X(1/\delta_n)$ s.t. $n\delta_n \to \infty$ as $n \to \infty$.

ExceedGAN: Extrapolation principle

Take advantage of $U_X(t) = t^\gamma L(t)$ ($U_X \in \mathcal{RV}_\gamma$) to link the extreme quantile

$$q_{Y(\delta_n)}(1-z) = q_X(1-\delta_n z) = U_X(1/(\delta_n z)),$$

and the threshold $q_X(1-\delta_n) = U_X(1/\delta_n)$ s.t. $n\delta_n \to \infty$ as $n \to \infty$.

Idea. Introduce the log-spacing function,

$$\begin{aligned}\log U_X(1/(\delta_n z)) - \log U_X(1/\delta_n) &= \gamma \log(1/z) + \varphi\big(\log(1/z), \log(1/\delta_n)\big) \\ &=: f\big(\log(1/z), \log(1/\delta_n)\big)\end{aligned}$$

with

$$(x_1, x_2 > 0) \mapsto \varphi(x_1, x_2) := \log\left(\frac{L(\exp(x_1 + x_2))}{L(\exp(x_2))}\right)$$

ExceedGAN: Extrapolation principle

Take advantage of $U_X(t) = t^\gamma L(t)$ ($U_X \in \mathcal{RV}_\gamma$) to link the extreme quantile

$$q_{Y(\delta_n)}(1-z) = q_X(1-\delta_n z) = U_X(1/(\delta_n z)),$$

and the threshold $q_X(1-\delta_n) = U_X(1/\delta_n)$ s.t. $n\delta_n \to \infty$ as $n \to \infty$.

Idea. Introduce the log-spacing function,

$$\begin{aligned}\log U_X(1/(\delta_n z)) - \log U_X(1/\delta_n) &= \gamma \log(1/z) + \varphi\big(\log(1/z), \log(1/\delta_n)\big) \\ &=: f\big(\log(1/z), \log(1/\delta_n)\big)\end{aligned}$$

with

$$(x_1, x_2 > 0) \mapsto \varphi(x_1, x_2) := \log\left(\frac{L(\exp(x_1 + x_2))}{L(\exp(x_2))}\right)$$

Unknown quantities.

1. Intermediate quantile $U_X(1/\delta_n)$
2. Tail index γ
3. Log-spacing function $\varphi(\cdot, \cdot)$

ExceedGAN: Extrapolation principle

Take advantage of $U_X(t) = t^\gamma L(t)$ ($U_X \in \mathcal{RV}_\gamma$) to link the extreme quantile

$$q_{Y(\delta_n)}(1-z) = q_X(1-\delta_n z) = U_X(1/(\delta_n z)),$$

and the threshold $q_X(1-\delta_n) = U_X(1/\delta_n)$ s.t. $n\delta_n \to \infty$ as $n \to \infty$.

Idea. Introduce the log-spacing function,

$$\begin{aligned}\log U_X(1/(\delta_n z)) - \log U_X(1/\delta_n) &= \gamma \log(1/z) + \varphi\big(\log(1/z), \log(1/\delta_n)\big) \\ &=: f\big(\log(1/z), \log(1/\delta_n)\big)\end{aligned}$$

with

$$(x_1, x_2 > 0) \mapsto \varphi(x_1, x_2) := \log\left(\frac{L(\exp(x_1+x_2))}{L(\exp(x_2))}\right)$$

Unknown quantities.

1. Intermediate quantile $U_X(1/\delta_n)$
2. Tail index γ
3. Log-spacing function $\varphi(\cdot,\cdot)$

Weissman. [Weissman, 1978]

1. $X_{n-k+1,n}$, $k = \lfloor n\delta_n \rfloor$
2. $\hat{\gamma}(k)$ [Hill, 1975]
3. 0

ExceedGAN: Bias correction (second order)

Second order condition. There exist $\gamma > 0, \rho_2 \leq 0$ and a function A_2 with $A_2(t) \to 0$ as $t \to \infty$ s.t. for all $z \geq 1$

$$\log\left(\frac{L(yt)}{L(t)}\right) = A_2(t) \int_1^y y_2^{\rho_2 - 1}\,\mathrm{d}y_2 + o(A_2(t)), \qquad \text{as } t \to \infty$$

ExceedGAN: Bias correction (second order)

Second order condition. There exist $\gamma > 0, \rho_2 \leq 0$ and a function A_2 with $A_2(t) \to 0$ as $t \to \infty$ s.t. for all $z \geq 1$

$$\log\left(\frac{L(yt)}{L(t)}\right) = A_2(t)\int_1^y y_2^{\rho_2 - 1}\,\mathrm{d}y_2 + o(A_2(t)), \qquad \text{as } t \to \infty$$

Ignoring the $o(\cdot)$ term and assuming (Hall-Welsh model)

$$A_2(t) = c_2 t^{\rho_2}$$

with $c_2 \neq 0$ and $\rho_2 < 0$, give a parametric approximation of $\varphi(x_1, x_2)$ as

$$\varphi^{\mathtt{NN}_J}(x_1, x_2; \theta) = c_2 \exp(\rho_2 x_2)(\exp(\rho_2 x_1) - 1)/\rho_2$$

$$= c_2\Big(\sigma^{\mathtt{E}}\left(\rho_2\left(x_1 + x_2\right)\right) - \sigma^{\mathtt{E}}\left(\rho_2 x_2\right)\Big)/\rho_2,$$

with $\theta = (\rho_2, c_2)$ and where $\sigma^{\mathtt{E}}(x) = \mathbb{I}_{\{x \geq 0\}}x + \mathbb{I}_{\{x<0\}}(\exp(x) - 1)$ is the **eLU** function.

ExceedGAN: Bias correction (J-th order)

J-th order condition. There exist $\gamma > 0$, and $\forall j \in \{2, \ldots, J\}$, $\rho_j \leq 0$ and functions A_j with $A_j(t) \to 0$ as $t \to \infty$ s.t. for all $y \geq 1$

$$\log\left(\frac{L(yt)}{L(t)}\right) = \sum_{j=2}^{J} \prod_{\ell=2}^{j} A_\ell(t) R_j(z) + o\left(\prod_{j=2}^{J} A_j(t)\right) \quad \text{as } t \to \infty, \tag{4}$$

$$R_j(z) = \int_1^y y_2^{\rho_2 - 1} \int_1^{y_2} y_3^{\rho_3 - 1} \cdots \int_1^{y_{j-1}} y_j^{\rho_j - 1} \, \mathrm{d}y_j \ldots \mathrm{d}y_3 \, \mathrm{d}y_2.$$

ExceedGAN: Bias correction (J-th order)

J-th order condition. There exist $\gamma > 0$, and $\forall j \in \{2, \dots, J\}$, $\rho_j \le 0$ and functions A_j with $A_j(t) \to 0$ as $t \to \infty$ s.t. for all $y \ge 1$

$$\log\left(\frac{L(yt)}{L(t)}\right) = \sum_{j=2}^{J} \prod_{\ell=2}^{j} A_\ell(t) R_j(z) + o\left(\prod_{j=2}^{J} A_j(t)\right) \quad \text{as } t \to \infty, \qquad (4)$$

$$R_j(z) = \int_1^y y_2^{\rho_2 - 1} \int_1^{y_2} y_3^{\rho_3 - 1} \cdots \int_1^{y_{j-1}} y_j^{\rho_j - 1} \,\mathrm{d}y_j \dots \mathrm{d}y_3 \,\mathrm{d}y_2.$$

Proposition (Approximation with $J(J-1)$ neurons [Allouche et al., 2024])

Assume the J-th order condition holds with $A_j(t) = c_j t^{\rho_j}$, where $c_j \neq 0$ and $\rho_j < 0$ for $j \in \{2, \dots, J\}$. Then, for all $x_1 > 0$ and $x_2 > 0$

$$\varphi(x_1, x_2) = \sum_{i=1}^{J(J-1)/2} w_i^{(1)} \left(\sigma^{\mathtt{E}}\left(w_i^{(2)} x_1 + w_i^{(3)} x_2\right) - \sigma^{\mathtt{E}}(w_i^{(4)} x_2)\right) + o\,(\dots)$$

with $w_i^{(1)} \in \mathbb{R}$, $w_i^{(2)} < 0$, $w_i^{(3)} < 0$, $w_i^{(4)} < 0$, $\forall i \in \{1, \dots, J(J-1)/2\}$.

ExceedGAN: Main Result

Log-spacing.

$$\log q_{Y(\delta_n)}(1-z) - \log q_X(1-\delta_n) = \gamma \log(1/z) + \varphi\big(\log(1/z), \log(1/\delta_n)\big)$$
$$= f(\log(1/z), \log(1/\delta_n))$$

Neural Network approximation.

$$q^{\mathtt{NN}_J}_{Y(\delta_n)}(1-z;\phi) := q_X(1-\delta_n)\,\exp\left(f^{\mathtt{NN}_J}(\log(1/z), \log(1/\delta_n);\phi)\right) \tag{5}$$

where $f^{\mathtt{NN}_J}(x_1, x_2;\phi) := w_0 x_1 + \varphi^{\mathtt{NN}_J}(x_1, x_2;\theta)$, with $\phi := (w_0, \theta)$.

ExceedGAN: Main Result

Log-spacing.

$$\begin{aligned}\log q_{Y(\delta_n)}(1-z) - \log q_X(1-\delta_n) &= \gamma \log(1/z) + \varphi\big(\log(1/z), \log(1/\delta_n)\big)\\ &= f(\log(1/z), \log(1/\delta_n))\end{aligned}$$

Neural Network approximation.

$$q^{\mathrm{NN}_J}_{Y(\delta_n)}(1-z;\phi) := q_X(1-\delta_n)\, \exp\left(f^{\mathrm{NN}_J}(\log(1/z), \log(1/\delta_n); \phi)\right) \qquad (5)$$

where $f^{\mathrm{NN}_J}(x_1, x_2; \phi) := w_0 x_1 + \varphi^{\mathrm{NN}_J}(x_1, x_2; \theta)$, with $\phi := (w_0, \theta)$.

Theorem

Assume the J-th order conditions of the Proposition 2 hold. Then, there exists a parameter by $\theta^\star$ and a threshold $t_0 \in (0,1)$ such that the **one hidden-layer** *NN (5) with $J(J-1)$ neurons verifies*

$$\sup_{z\in(0,1]} \left|\log q_{Y(\delta_n)}(1-z) - \log q^{\mathrm{NN}_J}_{Y(\delta_n)}(1-z;\phi^\star)\right| \le |\bar\rho_J \bar c_J|\, \delta_n^{-\bar\rho_J},$$

for all $0 < \delta_n \le t_0$, and where $\bar c_J = c_2 \times \cdots \times c_J$, $\bar\rho_J = \rho_2 + \cdots + \rho_J$.

ExceedGAN: Architecture

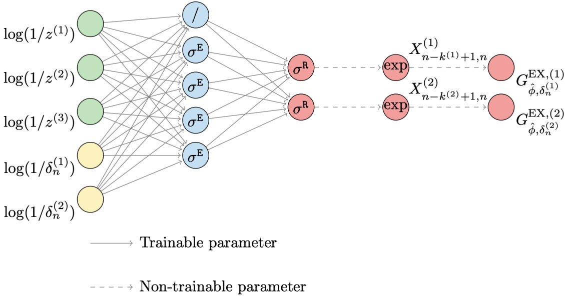


Figure: **Generator of the ExceedGAN with $d_z = 3$ and $d = 2$**

Versions.

1. **Fixed-level**: Plug the ExceedGAN generator into the GAN optimization problem for fixed levels $\delta_n^{(m)}, m \in \{1, \dots, d\}$
2. **Level-varying**: Conditional extansion method with adapted optimization problem and learning algorithm, see [Allouche et al., 2026, Section 3.2]

$$\arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \max_{\boldsymbol{\phi}\in\boldsymbol{\Phi}} \left(\mathsf{E}_{p_{\mathbf{U}}}\left\{\mathsf{E}_{p_{\mathbf{Y}(\mathbf{u}_n)}}\left\{\log D^{\mathrm{EX}}_{\boldsymbol{\phi}}(\mathbf{Y}, \boldsymbol{\delta}_n)\right\}\right\} + \mathsf{E}_{p_{\mathbf{U}}}\left\{\mathsf{E}_{p_{\mathbf{Z}}}\left\{\log\left[1 - D^{\mathrm{EX}}_{\boldsymbol{\phi}}\left\{G^{\mathrm{EX}}_{\boldsymbol{\theta}}(\mathbf{Z}, \boldsymbol{\delta}_n), \boldsymbol{\delta}_n\right\}\right]\right\}\right\}\right)$$

ExceedGAN: Validation on simulations

Data: Exceedances are simulated in the (bivariate) upper orthant

$$Q(\delta_n) := \left\{ \mathbf{x} \in \mathbb{R}^2 \mid x^{(1)} > q_{X^{(1)}}(1 - \delta_n^{(1)}), x^{(2)} > q_{X^{(2)}}(1 - \delta_n^{(2)}) \right\}$$

from a Gumbel copula with a dependence parameter $\mu \in \{1.1, 2, 10\}$ and Burr margins $(\gamma, \rho^{(1)})$, $(\gamma, \rho^{(2)})$ where $\gamma \in \{0.1, 0.5, 0.9\}$ and $(\rho^{(1)}, \rho^{(2)}) \in \{(-1, -2), (-1, -3), (-2, -3)\}$ using a basic acceptance-rejection method.

GAN, EV-GAN, (fixed-level) ExceedGAN.

1. $\delta_n = (0.1, 0.1)^\top$ with $1,000$ training and $10,000$ testing points.
2. $\delta_n = (0.05, 0.05)^\top$ with 250 training and $10,000$ testing points.

Level-varying ExceedGAN.

- Adaptation of the GAN algorithm to learn $Y(\delta_n) = X \mid X > q_X(1 - \delta_n)$ conditionally on $\delta_n \in [0, 0.5]^2$.
- 25.000 training and $10,000$ testing points.

ExceedGAN: Validation on simulations

▷ **Performance:** Mean square logarithmic error

▷ **Results:**

- FL ExceedGAN outperforms GAN, EV-GAN (44/54)
- LV ExceedGAN particularly efficient for $\gamma \geq 0.5$ in setting (2)

MSLE ($\boldsymbol{\delta}_n = (0.1, 0.1)^\top$)

		$\mu = 1.1$				$\mu = 2$				$\mu = 10$			
γ	(ρ_1, ρ_2)	GAN	EV-GAN	ExceedGAN		GAN	EV-GAN	ExceedGAN		GAN	EV-GAN	ExceedGAN	
0.3	$(-1,-2)$	0.46	0.52	**0.30**	0.38	0.23	0.26	**0.11**	0.46	0.34	0.97	0.18	**0.16**
	$(-1,-3)$	0.98	0.36	**0.35**	3.28	0.43	0.24	**0.13**	0.31	0.44	0.84	0.22	**0.20**
	$(-2,-3)$	0.53	0.35	0.39	**0.24**	0.14	0.24	**0.11**	0.76	0.29	1.29	**0.05**	0.12
0.5	$(-1,-2)$	4.59	1.49	1.07	**1.03**	3.18	1.11	**0.54**	6.72	2.64	1.33	**0.39**	1.01
	$(-1,-3)$	3.93	1.21	1.19	**1.16**	3.98	1.41	1.30	**1.19**	2.38	1.50	**0.22**	2.60
	$(-2,-3)$	6.03	**1.09**	1.67	2.45	1.22	3.05	**0.69**	2.53	1.93	1.30	**0.51**	2.09
0.9	$(-1,-2)$	61.92	**6.13**	6.68	13.54	35.82	3.77	**2.11**	15.72	28.88	4.83	**1.32**	86.05
	$(-1,-3)$	38.65	**4.50**	4.85	25.65	44.17	3.82	**2.59**	13.13	24.56	5.69	**1.20**	84.16
	$(-2,-3)$	41.19	5.49	**5.05**	21.92	45.03	4.11	**1.86**	15.15	36.71	3.73	**1.56**	83.37

MSLE ($\boldsymbol{\delta}_n = (0.05, 0.05)^\top$)

		$\mu = 1.1$				$\mu = 2$				$\mu = 10$			
γ	(ρ_1, ρ_2)	GAN	EV-GAN	ExceedGAN		GAN	EV-GAN	ExceedGAN		GAN	EV-GAN	ExceedGAN	
0.3	$(-1,-2)$	2.98	1.38	1.14	**0.36**	2.10	0.86	1.91	**0.36**	1.23	1.90	1.22	**0.62**
	$(-1,-3)$	2.75	2.76	**0.56**	2.60	2.35	1.28	1.50	**0.89**	1.35	1.82	1.27	**0.34**
	$(-2,-3)$	2.34	1.12	2.10	**0.17**	1.88	1.39	1.47	**0.46**	0.75	1.88	1.19	**0.43**
0.5	$(-1,-2)$	18.58	6.19	2.21	**1.18**	13.88	5.22	**2.21**	6.64	8.87	3.76	2.77	**1.60**
	$(-1,-3)$	15.96	4.71	5.55	**1.57**	15.08	3.20	**2.17**	2.40	9.06	4.37	**2.64**	3.16
	$(-2,-3)$	20.91	7.16	5.59	**1.11**	14.23	4.08	**2.16**	3.93	6.24	5.04	**2.18**	5.28
0.9	$(-1,-2)$	–	16.38	13.70	**12.97**	–	9.27	**7.81**	15.72	90.41	15.66	**7.10**	88.78
	$(-1,-3)$	–	22.34	17.54	**14.21**	–	12.06	**6.82**	12.91	–	15.62	**7.37**	86.84
	$(-2,-3)$	–	20.21	**14.31**	18.13	–	11.90	**9.96**	20.70	–	34.13	**7.46**	82.66

$\delta_n = (0.1, 0.1)^\top$, $\mu = 2$, $(\rho_1, \rho_2) = (-1, -3)$

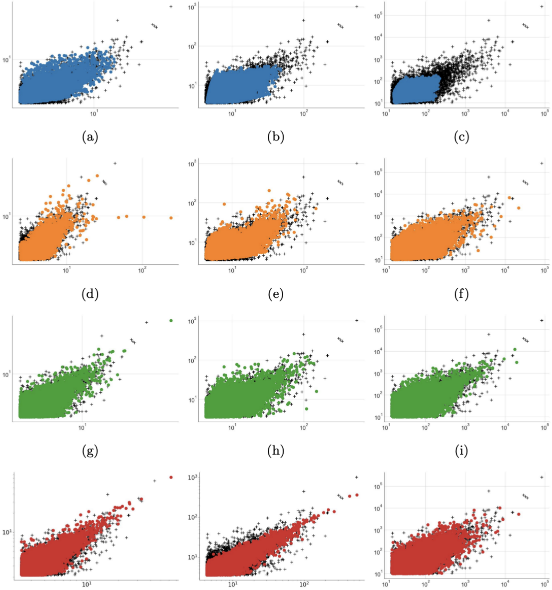
$\gamma = 0.3$ | $\gamma = 0.5$ | $\gamma = 0.9$

GAN

EV-GAN

FL-ExceedGAN

LV-ExceedGAN



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

$\delta_n = (0.05, 0.05)^\top$, $\mu = 2$, $(\rho_1, \rho_2) = (-1, -3)$

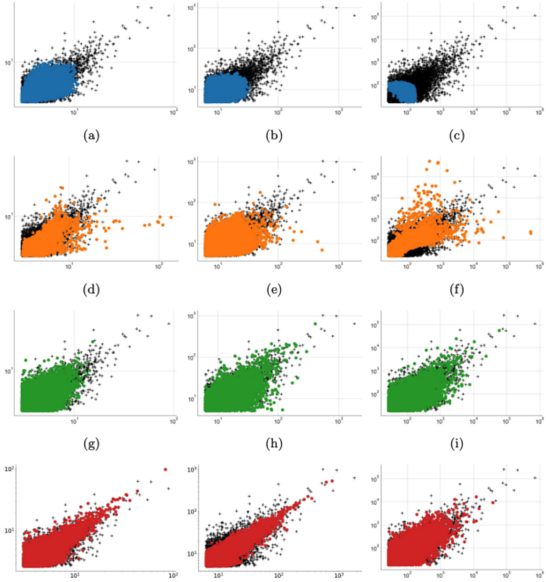
$\gamma = 0.3$ | $\gamma = 0.5$ | $\gamma = 0.9$

GAN

EV-GAN

FL-ExceedGAN

LV-ExceedGAN



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Application to crypto data - Risk Metrics

▷ Consider negative daily log-returns of BTC/USD and ETH/USD during 8 years (1116 observations) with $\hat{\gamma}_{\mathtt{btc}} \approx \hat{\gamma}_{\mathtt{eth}} \approx 0.32$ and $\hat{\mu} \approx 2.4$.

▷ Focus on the estimation of the Expected Shortfall, for $m \in \{1, 2\}$

$$\mathrm{ES}^{(m)}(1-\delta_n) = \frac{1}{\delta_n^{(m)}} \int_0^{\delta_n^{(m)}} F_{X^{(m)}}^{-1}(1-u)\,\mathrm{d}u = \int_0^1 q_{Y^{(m)}(\delta_n^{(m)})}(1-z)\,\mathrm{d}z.$$

▷ Estimation at levels $\delta_n = (0.1, 0.1)^\top$ (72 observations) and $\delta_n = (0.05, 0.05)^\top$ (31 observations)

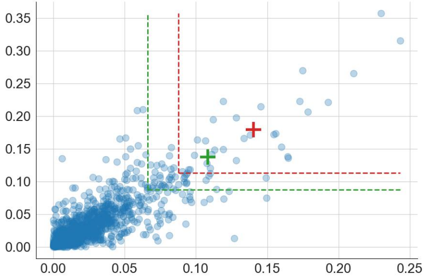


Figure: Scatter plot of the data. The empirical (component-wise) Expected Shortfall is depicted by a plus sign

Simulated Expected Shortfalls

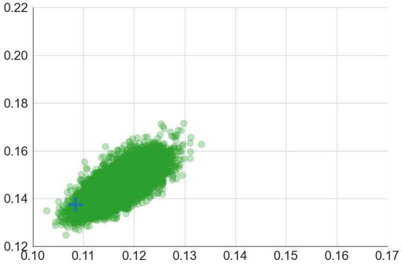


Figure: **Simulated Expected Shortfalls by level-varying ExceedGAN and empirical Expected Shortfall (blue plus sign) at levels $\delta_n = (0.1, 0.1)^\top$ (a) and $\delta_n = (0.05, 0.05)^\top$ (b) for the pairs BTC/USD (x-axis) and ETH/USD (y-axis)**

- **Classical bootstrap**: unsuitable for estimating means in heavy-tailed settings.
- **Proposed method**: take into account unseen points without assumptions on the underlying distribution.

4. Conclusion

- **Vanilla GAN**: not able to reproduce heavy-tailed events, leading to a severe **under-estimation of extreme risks**.
- **EV-GAN**: new parametrization of a **generative model** dedicated to **extreme events**.
- **ExceedGAN**: Improve EV-GAN results on margins using (refined) **extrapolation methods**.
- Further work: Combine the results of [Girard et al., 2026] (showing mathematically how the **extreme dependence structure** can be preserved) with the previous ones on the margins.

References I

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

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

Allouche, M., Girard, S., and Gobet, E. (2025).
Learning extreme expected shortfall and conditional tail moments with neural networks. application to cryptocurrency data.
Neural Networks, 182:106903.

References II

Allouche, M., Girard, S., and Gobet, E. (2026).
ExceedGAN: simulation above extreme thresholds using Generative Adversarial Networks.
Extremes, pages 1–23.

Bhatia, S., Jain, A., and Hooi, B. (2020).
ExGAN: Adversarial generation of extreme samples.
arXiv preprint arXiv:2009.08454.

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

References III

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

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

de Haan, L. and Ferreira, A. (2006).
Extreme value theory.
Springer Series in Operations Research and Financial Engineering.
Springer, New York.

References IV

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

Girard, S., Gobet, E., and Pachebat, J. (2026+).
HTGAN: Heavy-tail GAN for multivariate dependent extremes via latent-dimensional control.
International Journal of Computer Mathematics.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014).
Generative adversarial nets.
In *Advances in neural information processing systems*, pages 2672–2680.

References V

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

Hill, B. M. (1975).
A simple general approach to inference about the tail of a distribution.
The Annals of Statistics, 3(5):1163–1174.

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

Lafon, N., Naveau, P., and Fablet, R. (2023).
A VAE approach to sample multivariate extremes.
arXiv preprint arXiv:2306.10987.

References VI

Lhaut, S., Rootzén, H., and Segers, J. (2026+).
Simulation of multivariate extremes: A Wasserstein–Aitchison GAN approach.
Extremes.

Maiorov, V. E. (1999).
On best approximation by ridge functions.
J. Approx. Theory, 99(1):68–94.

Pinkus, A. (1999).
Approximation theory of the MLP model in neural networks.
In *Acta numerica, 1999*, volume 8 of *Acta Numer.*, pages 143–195. Cambridge Univ. Press, Cambridge.

Villani, C. (2009).
Optimal transport, volume 338 of *Fundamental Principles of Mathematical Sciences*.
Springer-Verlag, Berlin.

References VII

Weissman, I. (1978).
Estimation of parameters and large quantiles based on the k largest observations.
Journal of the American Statistical Association, 73(364):812–815.

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