ExcessGAN: simulation above extreme thresholds using Generative Adversarial Networks

Michaël Allouche   Stéphane Girard   Emmanuel Gobet

Generative AI Modelling for Extreme Events
The University of Edinburgh

June 13-14, 2025

Motivations

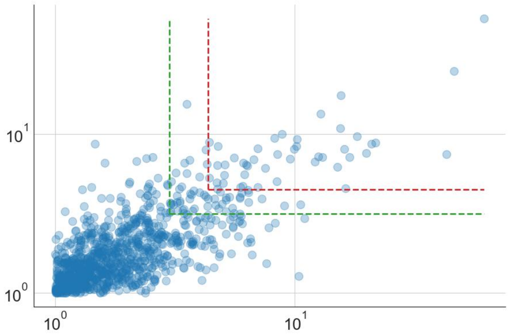


Figure: **Simulated observations from a bivariate Gumbel copula with two Burr margins. Two extreme regions are represented by green and red dashed lines.**

How to sample from the distribution of the excess?

- **Stochastic models:** high computational complexity
- **Data based models:** few observations by nature

Ingredients: Generative Modeling, Neural Networks, Extreme Value Theory

Generative modeling

Objective. Given observations $\{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ assumed to be indep. sampled from an unknown distribution $p_{\mathbf{X}}$ on $\mathcal{X} \subseteq \mathbb{R}^D$, find a generator $G : \mathcal{Z} \to \mathcal{X}$ and a latent distribution $p_{\mathbf{Z}}$ defined on $\mathcal{Z} \subseteq \mathbb{R}^q$, s.t.

$$G(\mathbf{Z}) \stackrel{\mathrm{d}}{=} \mathbf{X}, \quad \mathbf{Z} \sim p_{\mathbf{Z}}.$$

Generative modeling

Objective. Given observations $\{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ assumed to be indep. sampled from an unknown distribution $p_{\mathbf{X}}$ on $\mathcal{X} \subseteq \mathbb{R}^D$, find a generator $G : \mathcal{Z} \to \mathcal{X}$ and a latent distribution $p_{\mathbf{Z}}$ defined on $\mathcal{Z} \subseteq \mathbb{R}^q$, s.t.

$$G(\mathbf{Z}) \overset{\mathrm{d}}{=} \mathbf{X}, \quad \mathbf{Z} \sim p_{\mathbf{Z}}.$$

Which $p_{\mathbf{Z}}$? Which q? How to approximate G?

Generative modeling

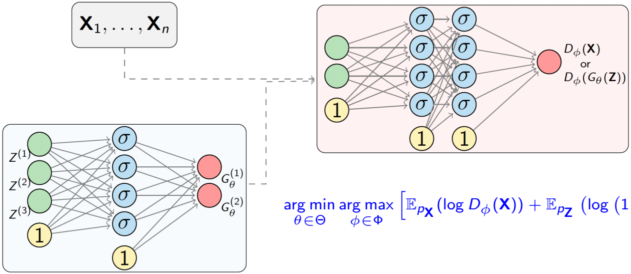
Objective. Given observations $\{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ assumed to be indep. sampled from an unknown distribution $p_{\mathbf{X}}$ on $\mathcal{X} \subseteq \mathbb{R}^D$, find a generator $G : \mathcal{Z} \to \mathcal{X}$ and a latent distribution $p_{\mathbf{Z}}$ defined on $\mathcal{Z} \subseteq \mathbb{R}^q$, s.t.

$$G(\mathbf{Z}) \stackrel{\mathrm{d}}{=} \mathbf{X}, \quad \mathbf{Z} \sim p_{\mathbf{Z}}.$$

Which $p_{\mathbf{Z}}$? Which q? How to approximate G?

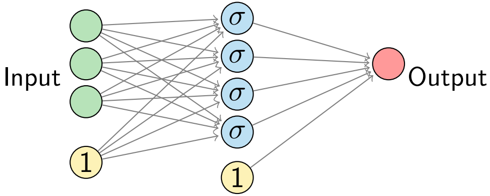
GANs. [Goodfellow et al., 2014]

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



$$\underset{\theta \in \Theta}{\arg\min}\ \underset{\phi \in \Phi}{\arg\max} \left[\mathbb{E}_{p_{\mathbf{X}}}(\log D_\phi(\mathbf{X})) + \mathbb{E}_{p_{\mathbf{Z}}}\left(\log\left(1 - D_\phi\left(G_\theta(\mathbf{Z})\right)\right)\right) \right]$$

Neural Networks



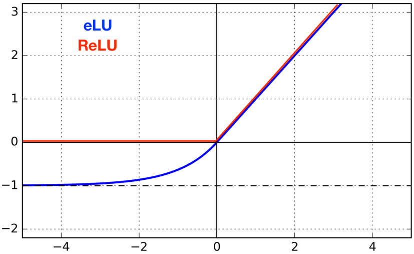


Figure: **Example of a one-hidden layer neural network with 4 neurons mapping from a 3 to a 1 dimensional space. The symbol σ represents the transformation with an activation function while the arrows stand for different parameters .**

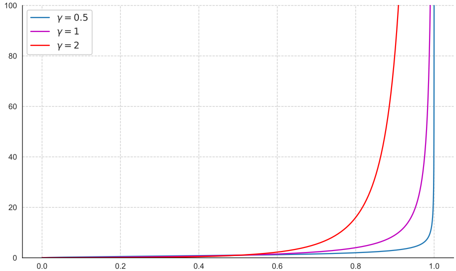
Theorem (Universal Approximation Theorem [Pinkus, 1999])

A one hidden-layer neural can **uniformly approximate on a compact set any continuous function** *with arbitrary precision as long as σ is not a polynomial.*

Extreme-value theory (in dimension 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.*

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

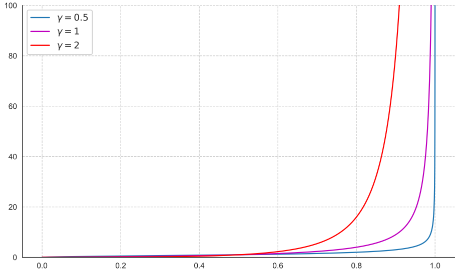


Quantile function of a Burr distribution $u \mapsto q(u)$ with parameters $\gamma = \{0.5, 1, 2\}$ and $\rho = -1$

Extreme-value theory (in dimension 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 $G_\theta(\mathbf{Z}) \stackrel{\mathrm{d}}{=} X$

Statistical framework

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

Statistical framework

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

Lemma

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

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

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

Statistical framework

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

Lemma

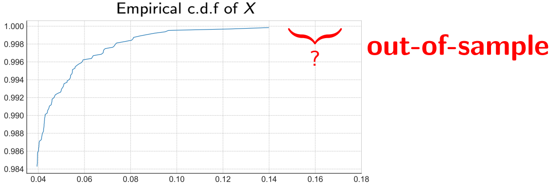
Assume that the distribution of X is continuous. Then, for all $\delta_n \in (0,1)$

$$Y(\delta_n) \overset{\mathrm{d}}{=} q_{Y(\delta_n)}(1-Z) = F_X^{-1}(1-\delta_n Z),$$

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

⚠ Challenge ⚠

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



Extrapolation principle

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

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

and the threshold $F_X^{-1}(1-\delta_n) = U_X(1/\delta_n)$.

Extrapolation principle

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

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

and the threshold $F_X^{-1}(1-\delta_n) = U_X(1/\delta_n)$.

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

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

Extrapolation principle

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

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

and the threshold $F_X^{-1}(1-\delta_n) = U_X(1/\delta_n)$.

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 $\varphi(\cdot,\cdot)$

Extrapolation principle

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

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

and the threshold $F_X^{-1}(1-\delta_n) = U_X(1/\delta_n)$.

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

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 $z \geq 1$

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

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

$$\begin{aligned}\varphi^{\mathtt{NN}_J}(x_1, x_2; \theta) &= c_2 \exp(\rho_2 x_2)(\exp(\rho_2 x_1) - 1)/\rho_2 \\ &= c_2\Big(\sigma^{\mathtt{E}}\left(\rho_2\left(x_1 + x_2\right)\right) - \sigma^{\mathtt{E}}\left(\rho_2 x_2\right)\Big)/\rho_2,\end{aligned}$$

with $\theta = (\rho_2, c_2)$ and where $\sigma^{\mathtt{E}}(x) = \mathbb{1}_{\{x \geq 0\}} x + \mathbb{1}_{\{x < 0\}}(\exp(x) - 1)$ is the **eLU** function, see [Allouche et al., 2024].

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 $y \geq 1$

$$\log\left(\frac{L(yt)}{L(t)}\right) = \sum_{j=2}^{J} \prod_{\ell=2}^{j} A_\ell(t) R_j(z) + o\left(\prod_{j=2}^{J} A_j(t)\right) \quad \text{as } t \to \infty, \tag{1}$$

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

Proposition

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 $j \in \{2, \ldots, J\}$. Then, for all $x_1 > 0$ and $x_2 > 0$

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

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

Results

Neural Network approximation.

$$q^{\mathtt{NN}_J}_{Y(\delta_n)}(1-z;\phi) := F_X^{-1}(1-\delta_n)\,\exp\left(f^{\mathtt{NN}_J}(\log(1/z),\log(1/\delta_n);\phi)\right) \qquad (2)$$

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

Results

Neural Network approximation.

$$q^{\mathrm{NN}_J}_{Y(\delta_n)}(1-z;\phi) := F_X^{-1}(1-\delta_n)\exp\left(f^{\mathrm{NN}_J}(\log(1/z),\log(1/\delta_n);\phi)\right) \quad (2)$$

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 1 hold. Then, there exists a parameter by $\theta^\star$ and a threshold $t_0 \in (0,1)$ such that the **one hidden-layer** *NN (2) 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 $\bar{c}_J = c_2\times\cdots\times c_J$, $\bar{\rho}_J = \rho_2+\cdots+\rho_J$.

ExcessGAN

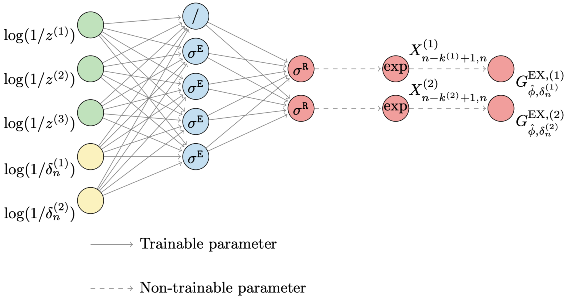


Figure: **Generator of the ExcessGAN with $q = 3$ and $D = 2$**

Versions.

1. **Fixed-level**: Plug the ExcessGAN generator into the GAN optimization problem for fixed levels $\delta_n^{(m)}, m \in \{1, \ldots, D\}$
2. **Level-varying**: Conditional extansion method with adapted optimization problem and learning algorithm, see [Allouche et al., 2025, Section 3.2]

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

Pseudo-code

Algorithm 1: Level-varying ExcessGAN training

Input: m_U, *batch size of the conditional variable*,
m_X, batch size of the data for each conditional value
a, left support of the conditional distribution

Output: $(\hat{\theta}, \hat{\phi})$, *trained parameters*

for *number of iterations* **do**

sample a minibatch of m_U levels $\{\delta_k^{(1)} \sim \mathcal{U}(a,1), \ldots, \delta_k^{(D)} \sim \mathcal{U}(a,1)\}_{k=1}^{m_U}$
and store the associated thresholds $\{u_k^{(1)}, \ldots, u_k^{(D)}\}_{k=1}^{m_U}$

for $k = 1 : m_U$ **do**

sample a minibatch of m_X data $\{\mathbf{x}_{i,k} \in Q(\boldsymbol{\delta}_k)\}_{i=1}^{m_X}$ with replacement

Update the discriminator by computing the gradient

$$\nabla_\phi \frac{1}{m_X m_U} \sum_{k=1}^{m_U} \sum_{i=1}^{m_X} \left[\log(D_{\boldsymbol{\phi}}^{\mathrm{EX}}(\mathbf{x}_{i,k}, \boldsymbol{\delta}_k)) + \log(1 - D_{\boldsymbol{\phi}}^{\mathrm{EX}}(G_{\boldsymbol{\theta}}^{\mathrm{EX}}(\mathbf{z}_{i,k}, \boldsymbol{\delta}_k), \boldsymbol{\delta}_k))\right],$$

with $\mathbf{z}_{i,k} \sim p_{\mathbf{Z}}$.

Update the generator by computing the gradient

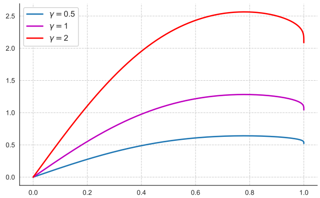
$$\nabla_{\boldsymbol{\theta}} \frac{1}{m_X m_U} \sum_{k=1}^{m_U} \sum_{i=1}^{m_X} \left[\log(1 - D_{\boldsymbol{\phi}}^{\mathrm{EX}}(G_{\boldsymbol{\theta}}^{\mathrm{EX}}(\mathbf{z}_{i,k}, \boldsymbol{\delta}_k), \boldsymbol{\delta}_k))\right],$$

with $\mathbf{z}_{i,k} \sim p_{\mathbf{Z}}$.

Competitor: EV-GAN [Allouche et al., 2022]

1. Find a Tail-index function (**TIF**) f^{TIF} **continuous** and **bounded** on $[0,1]$ for all **heavy-tailed** distributions s.t. $f^{\mathrm{TIF}}(u) \to \gamma$ as $u \to 1$

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

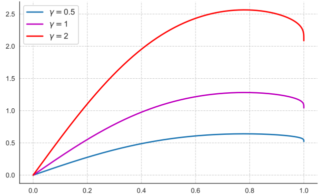


TIF associated with Burr distribution $(\rho = -1)$

Competitor: EV-GAN [Allouche et al., 2022]

1. Find a Tail-index function (**TIF**) f^{TIF} **continuous** and **bounded** on $[0,1]$ for all **heavy-tailed** distributions s.t. $f^{\mathrm{TIF}}(u) \to \gamma$ as $u \to 1$

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

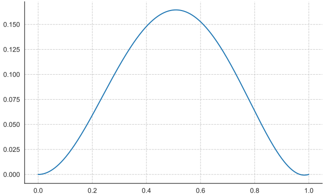


TIF associated with Burr distribution $(\rho = -1)$

2. For better approximation, find a Correction TIF (**CTIF**)

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

which enjoys higher regularity around $u = 1$



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

Experiments - Simulated data

We simulate excess in the upper orthant

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

from a Gumbel Copula with a dependence parameter μ and Burr margins (γ, ρ) using basic acceptance-rejection method - **54 configurations**.

GAN, EV-GAN, Fixed-level ExcessGAN.

1. $\delta_n = (0.1, 0.1)^\top$ with 1000 training and 10K testing points
2. $\delta_n = (0.05, 0.05)^\top$ with 250 training and 10K testing points

Level-varying ExcessGAN.

- $\delta_n \in [0.5, 1]^2$
- 100K training points and same testing sets as above

▷ **Performance:** Mean square logarithmic error

▷ **Results:**

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

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

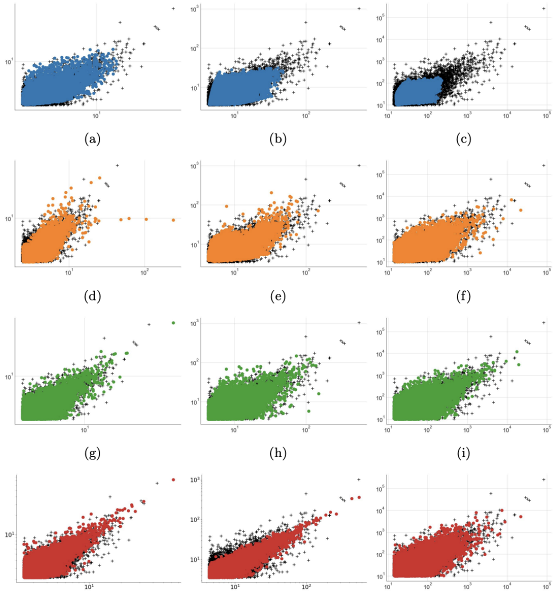
$\gamma = 0.3$ | $\gamma = 0.5$ | $\gamma = 0.9$

GAN

EV-GAN

FL-ExcessGAN

LV-ExcessGAN



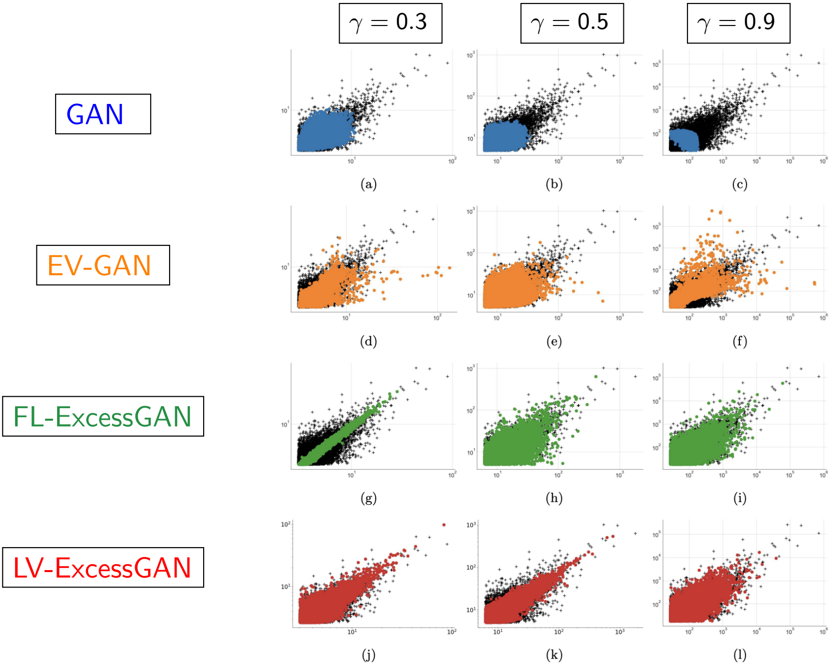
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

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



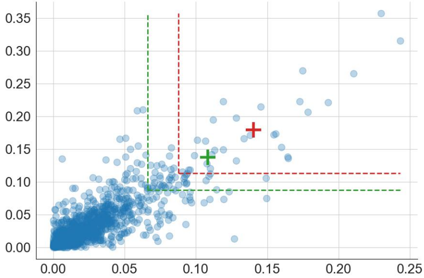
Application to crypto data - Risk Metrics

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



Simulated Expected Shortfalls

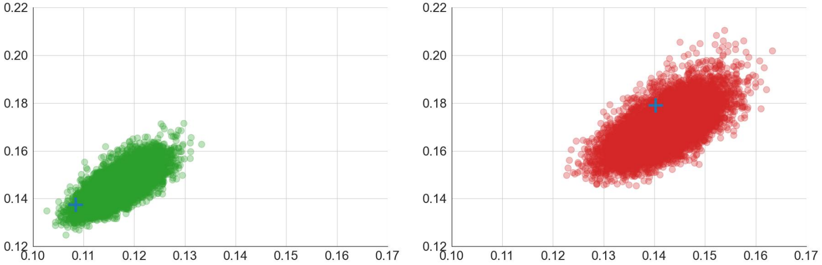


Figure: **Simulated Expected Shortfalls by level-varying ExcessGAN and empirical Expected Shortfall (blue plus sign) at levels $\delta_n = (0.1, 0.1)^\top$ (a) and $\delta_n = (0.05, 0.05)^\top$ (b) for the pairs BTC/USD (x-axis) and ETH/USD (y-axis)**

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

▷ **Next:** Compare with another neural network ES estimator [Allouche et al., 2023]

Conclusion

- **Generative model** dedicated to extremes which is able to learn the **distribution of the excess** considering a **bias reduction** technique through an appropriate **eLU basis functions**.
- **Dominance of the ExcessGAN** in a bunch of heavy-tailed cases, with a benefit to use the **FL version when enough observations** are available in the tails; but also with the **LV version as a competitive alternative** that performs well in **all regions**.
- Application for risk measure estimation using **data augmentaiton**

▷ **Next:** Address the extreme dependence structure

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. (2023).
Learning out-of-sample expected shortfall and Conditional Tail Moments with neural networks. application to cryptocurrency data.
Neural Networks, 60(62G32):68T07.

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

References II

Allouche, M., Girard, S., and Gobet, E. (2025).
ExcessGAN: simulation above extreme thresholds using generative adversarial networks.
Under review.

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.

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

References III

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

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.