GPDFlow: Generative Multivariate Threshold Exceedance Modeling via Normalizing Flows

Chenglei Hu and Daniela Castro-Camilo

University of Glasgow

c.hu.2@research.gla.ac.uk

Generative AI Modelling for Extreme Events - Workshop

14 June 2025

Challenges in Multivariate Extreme Exceedance Modelling

- Within the **asymptotic dependence (AD)** regime, the limiting dependence structure is **NOT uniquely defined**, leading to infinite parameterizations.
- Few tractable models for complex tail dependence in multivariate extremes.
- Propagation of Error in the two-step marginal and dependence estimation.
- Classical multivariate EVT typically targets **joint extremes**, but real-world applications often involve **partial extremes**, e.g.:

$$\mathbb{P}(X_1 < q_{1,0.5},\ X_2 > q_{2,0.99})$$

- Generative models are powerful tools for approximating distributions, but they often exhibit poor performance on heavy-tailed regions.

Overview

Goals of Our Work

- Address the **infinite parameterisations of the dependence function** in the AD setting.
- Develop a likelihood-based generative model that
  1. jointly estimates **marginal** distributions and tail **dependence**
  2. naturally accommodates **partial extreme events**, while retaining interpretable parameters.

Our Approach

Combine the **Multivariate Generalised Pareto Distribution (mGPD)** with generative models — specifically, **normalising flows**.

Threshold Exceedance Definitions

We focus on the threshold exceedance setting where **at least one component exceeds its threshold**.

This "reversed L-shaped" support may seem unusual, but it's highly practical for computing probabilities such as:

- $\mathbb{P}(X_1 > x_1, X_2 > x_2)$
- $\mathbb{P}(X_1 < u_1, X_2 > x_2)$

Multivariate Generalized Pareto Distribution

Assume $\boldsymbol{Y} \in \mathbb{R}^d$ has a joint distribution function F in the max-domain of attraction of a non-degenerate limit distribution G.

That is , there exists sequences $\boldsymbol{a}_n \in (0, \infty)^d$ and $\boldsymbol{b}_n \in \mathbb{R}^d$ such that

$$\lim_{n \to \infty} n\{1 - F(\boldsymbol{a}_n \boldsymbol{y} + \boldsymbol{b}_n)\} = -\log G(\boldsymbol{y}).$$

Then, for a high threshold vector $\boldsymbol{u}$, the conditional excess vector $\boldsymbol{Y} - \boldsymbol{u} | \boldsymbol{Y} \not\leq \boldsymbol{u}$ converges to the **multivariate generalised Pareto distribution (mGPD)**, with cdf

$$H(\boldsymbol{x}) = \frac{1}{\log G(\boldsymbol{0})} \log \frac{G(\boldsymbol{x} \wedge \boldsymbol{0})}{G(\boldsymbol{x})},$$

where $\wedge$ denotes the elementwise minimum.

Multivariate Generalized Pareto Distribution

Some appealing properties of the mGPD (Kiriliouk et al., 2019):

1. **Conditional Margins are Univariate GPDs:**

$$\mathbb{P}(X_j > x_j | X_j > 0) = \frac{1 - H_j(x_j)}{1 - H_j(0)} = \left(1 + \frac{\gamma_j x_j}{\sigma_j}\right)^{-1/\gamma_j}, \quad x_j > 0$$

2. **Threshold Stability:** Increasing the threshold by $\boldsymbol{\omega}$ yields an mGPD that only differs in marginal scale parameters, with new scale as $\boldsymbol{\sigma}_{\text{new}} = \boldsymbol{\sigma} + \boldsymbol{\gamma}\boldsymbol{\omega}$

Given that the conditional margins of the mGPD is univariate GPD, we can standardise the mGPD by

$$\boldsymbol{Z} = g_{\text{std}}(\boldsymbol{X}; \boldsymbol{\sigma}, \boldsymbol{\gamma}) = \mathbb{1}\{\boldsymbol{\gamma} \neq \boldsymbol{0}\} \frac{1}{\boldsymbol{\gamma}} \log\left(\frac{\boldsymbol{\gamma}\boldsymbol{X}}{\boldsymbol{\sigma}} + 1\right) + \mathbb{1}\{\boldsymbol{\gamma} = \boldsymbol{0}\} \frac{\boldsymbol{X}}{\boldsymbol{\sigma}},$$

so that $\boldsymbol{\gamma} = \boldsymbol{0}$, and $\mathbb{P}(Z_j > z_j | Z_j > 0) = \exp(-z_j)$ for $z_j > 0$.

We say that $\boldsymbol{Z}$ follows a **standardised mGPD** and denote its density as $h(\boldsymbol{z})$.

Standardised mGPD

- To derive the density $h(\boldsymbol{z})$, we leverage the stochastic representation of the mGPD.
- According to Rootzén et al. (2018), any random variable Z follows a standardized mGPD can be expressed as

$$\boldsymbol{Z} = E + \boldsymbol{T} - \max(\boldsymbol{T})$$

- Here, $E \sim \text{Exp}(1)$, $\boldsymbol{T}$ is a d-dimensional random vector independent to E and satisfying two weak conditions

  1. $\mathrm{P}(T_j > -\infty) > 0$
  2. $\mathrm{P}(\max(\boldsymbol{T}) > -\infty) = 1$

Standardised mGPD

Using the stochastic representation $\boldsymbol{Z} = E + \boldsymbol{T} - \max(\boldsymbol{T})$, the density of the standardised mGPD is:

$$h(\boldsymbol{z}) = \frac{\mathbb{1}\{\max(\boldsymbol{z} > 0)\}}{\exp\{\max(\boldsymbol{z})\}} \int_{-\infty}^{\infty} f_{\boldsymbol{T}}(\boldsymbol{z} + s)\mathrm{d}s,$$

where $f_{\boldsymbol{T}}$ is the density of $\boldsymbol{T}$.

This formulation admits **infinitely many parameterisations** via different choices of $\boldsymbol{T}$.

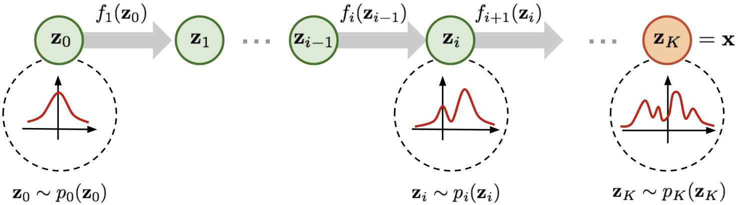
Existing approaches only explore $\boldsymbol{T}$ with closed-form $f_{\boldsymbol{T}}$ (e.g., independent Gumbel, reverse-Gumbel, reverse exponential).

Our Proposal: GPDflow

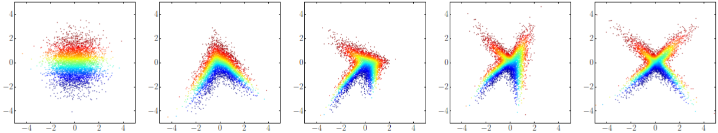
Use **normalising flows** to model $\boldsymbol{T}$ flexibly and efficiently!

Normalising Flows transform a simple base distribution into a complex target distribution using a sequence of invertible, differentiable mappings.

Core idea: Change of variables in probability density functions!



Two-dimensional example from Papamakarios et al. (2021):



Normalising Flows

- For any well-behaved target distribution, there exists an **invertible** and **differentiable** transformation from a simple base (e.g., Uniform or Gaussian; Papamakarios et al. 2021).

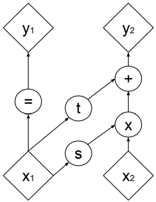
- Neural networks are flexible and differentiable, but **not naturally invertible**.

- **Normalising flows overcome this** by carefully designing network architectures to ensure invertibility and efficient Jacobian computation.

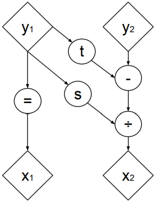
- Jacobian of the transformation should also be easy to compute.

- We use **Real NVP**, a simple but powerful flow model that constructs the mapping by affine transformations.

Real NVP (Dinh et al., 2017) uses affine coupling layers to construct invertible transformations.



(a) Forward propagation



(b) Inverse propagation

Mathematically,

$$
\begin{aligned}
y_{1:d} &= x_{1:d} \\
y_{d+1:D} &= x_{d+1:D} \odot \exp\left(s\left(x_{1:d}\right)\right) + t\left(x_{1:d}\right),
\end{aligned}
$$

where $s(\cdot)$ and $t(\cdot)$ are scale and shift functions, parameterised by neural networks (e.g. MLPs).

Jacobian of Real NVP

An appealing feature of such architecture is that the Jacobian of such a transformation is **lower-triangular**:

$$\frac{\partial y}{\partial x^T} = \left[\begin{array}{cc} \mathbb{I}_d & 0 \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}^T} & \operatorname{diag}\left(\exp\left[s\left(x_{1:d}\right)\right]\right) \end{array} \right].$$

Key implications:

- Determinant is simply $\prod \exp(s_j) = \exp\left(\sum s_j\right)$
- We can design s and t as complex networks—without compromising tractability.

Normalising Flows: Inference

Inference with normalising flows closely **resembles standard maximum likelihood estimation**

Let $\boldsymbol{\theta}$ denote the flow parameters (e.g., neural network weights). The density of the transformed variable is:

$$f_{\boldsymbol{Y}}(\boldsymbol{y} \mid \boldsymbol{\theta}) = f_{\boldsymbol{U}}(\boldsymbol{u}) \cdot \left|\det J_g(\boldsymbol{u} \mid \boldsymbol{\theta})\right|^{-1}, \quad \boldsymbol{u} = g^{-1}(\boldsymbol{y} \mid \boldsymbol{\theta})$$

Why this matters:

- We obtain the exact likelihood, not just an approximation.
- Enables principled training via maximum likelihood.
- A key feature over GANs or Diffusion Models!

GPDFlow

Key Idea: Represent the latent variable $\boldsymbol{T}$ in the stochastic representation of the standardised mGPD using a normalising flow (Real NVP).

Formally, GPDFlow is a d-dimensional distribution with density

$$f(\boldsymbol{x};\boldsymbol{\sigma},\gamma,\boldsymbol{\theta}) = \frac{\mathbb{1}\{\max(\boldsymbol{z})>0\}}{\exp\{\max(\boldsymbol{z})\}} \int_{-\infty}^{\infty} f_{\boldsymbol{U}}\left(g^{-1}(\boldsymbol{z}+s;\boldsymbol{\theta})\right) \left|\det J_g(\boldsymbol{z}+s;\boldsymbol{\theta})\right|^{-1} ds \times \prod_{j=1}^{d} \frac{1}{\sigma_j+\gamma_j x_j}$$

where $\boldsymbol{z} = g_{\text{std}}(\boldsymbol{x})$ is the standardized version.

Two Perspectives:

1. **Statistical:** GPDFlow is an mGPD with highly flexible dependence structure.
2. **Machine Learning:** A sample from f is generated by:

$$\boldsymbol{X} = g_{\text{std}} \circ g_{\text{mGPD}} \circ g(\boldsymbol{U})$$

where $g_{\text{mGPD}}(\boldsymbol{T}) = E + \boldsymbol{T} - \max(\boldsymbol{T})$, and $E \sim \text{Exp}(1)$.

Advantage of the GPDFlow structure

1. Enhance the flexibility of the tail dependence
2. Avoid Real NVP directly modelling marginal tail, where Real NVP generally struggles to estimate tail heaviness accurately
3. Jointly estimate the margins and dependence
4. Full likelihood inference instead of censored likelihood, being able to model partial extremes.

Inference for GPDFlow

Let $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ be observed threshold exceedances. The log-likelihood is:

$$\ell(\boldsymbol{\sigma}, \gamma, \boldsymbol{\theta}) = \sum_{i=1}^{n} \left\{ \log \left[\int_{-\infty}^{\infty} f_{\boldsymbol{U}} \left(g^{-1}(\boldsymbol{z}_i + s; \boldsymbol{\theta}) \right) \left| \det J_g(\boldsymbol{z}_i + s; \boldsymbol{\theta}) \right|^{-1} ds \right] \right.$$
$$\left. + \log \mathbb{1}\{\max(\boldsymbol{z}_i) > 0\} - \max(\boldsymbol{z}_i) - \sum_{j=1}^{d} \log(\sigma_j + \gamma_j x_{ij}) \right\},$$

where $\boldsymbol{z}_i = g_{\text{std}}(\boldsymbol{x}_i)$.

Note: Although $\boldsymbol{z}_i$ is d-dimensional, the integral is only over a scalar variable s.

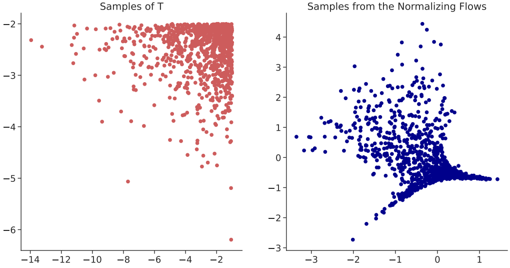
We update $(\boldsymbol{\theta}, \boldsymbol{\sigma}, \gamma)$ by maximize the **full likelihood** $\ell(\boldsymbol{\sigma}, \gamma, \boldsymbol{\theta})$.

Identifiability of $\boldsymbol{T}$

Recall the stochastic representation:

$$\boldsymbol{Z} = E + \boldsymbol{T} - \max(\boldsymbol{T})$$

For any d-dimensional random vector $\boldsymbol{R} = (R, \cdots, R)$, $\boldsymbol{T}$ and $\boldsymbol{T} + \boldsymbol{R}$ will always lead to same $\boldsymbol{Z}$.



$\boldsymbol{S} = \boldsymbol{T} - \max(T)$ is identifiable, but is applying $\max(S) = 0$ is difficult in normalising flows.

Fortunately, the identification issue **does NOT affect** the tail inference.

Threshold Selection in GPDFlow

General idea: use the threshold-stability property

We select the threshold q^* based on **stability of tail dependence measures**:

- Compute

$$\chi_{1:d}(q) = \frac{\mathbb{P}\{\bigcap_{j=1}^{d}\{X_j > F_j^{-1}(q)\}\}}{1-q}, \quad q \in (0,1) \leftarrow \text{ joint exceedance measure },$$

$$\omega_{1:d}(q) = \frac{\mathbb{P}\{\bigcup_{j=1}^{d}\{X_j > F_j^{-1}(q)\}\}}{1-q}, \quad q \in (0,1) \leftarrow \text{ union exceedance measure.}$$

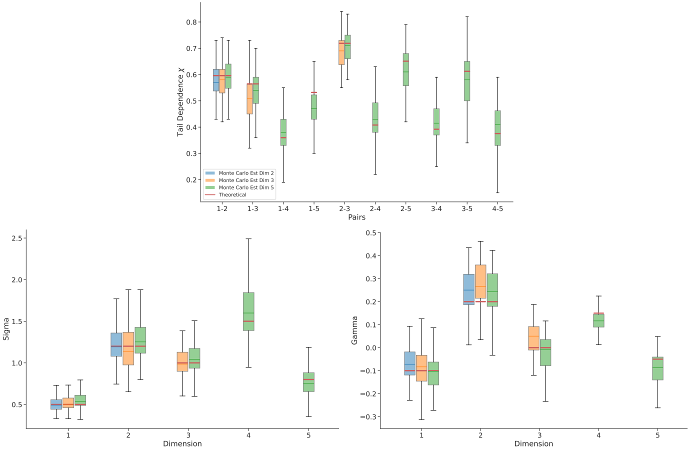
- Plot empirical $\hat{\chi}_{1:d}(q)$ and $\hat{\omega}_{1:d}(q)$
- Identify where both stabilise $\rightarrow q^* = \max\{q_\chi, q\omega\}$
- Final threshold: $(F_1^{-1}(q^*), \ldots, F_d^{-1}(q^*))$

This balances bias–variance trade-off and captures both joint and partial extremes reliably.

Simulation 1: Parameter Estimation Accuracy

We simulate from a regular parametric multivariate GPD (mGPD), and assess whether GPDFlow can recover:

- Pairwise tail dependence (χ)
- Marginal parameters ($\boldsymbol{\sigma}$ and $\boldsymbol{\gamma}$)



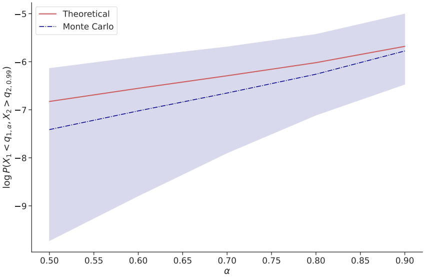
Simulation 2: Partial Exceedance Probability

We test GPDFlow's ability to estimate partial exceedance probabilities.

Setup:

- 1200 samples from a bivariate Gumbel copula ($\theta = 1.3$)
- Margins: Gaussian with $(\mu_1, s_1) = (1, 3)$ and $(\mu_2, s_2) = (2, 5)$
- Threshold: 0.95-quantile for each margin

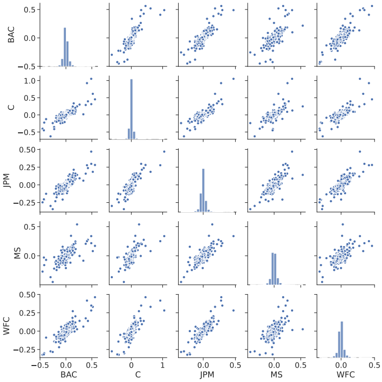
Examined metric: $\mathbb{P}(X_1 < q_{1,\alpha}, X_2 > q_{2,0.99})$

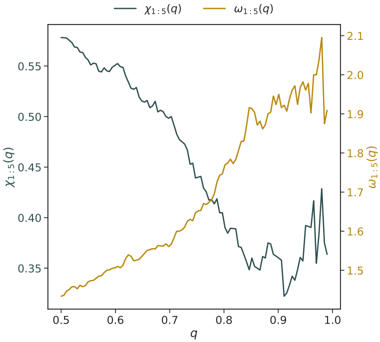


We analyse the 5-day negative log returns of 5 major US banks (2005–2025).

Risk measure: Conditional Value-at-Risk (CoVaR)

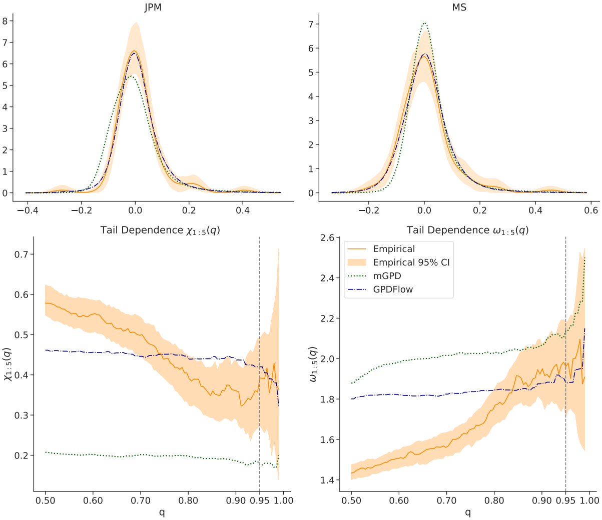
$$\mathrm{CoVaR}_{\alpha,\beta}(Y \mid X) = \mathrm{VaR}_{\alpha}\left(Y \mid X \geq \mathrm{VaR}_{\beta}(X)\right)$$



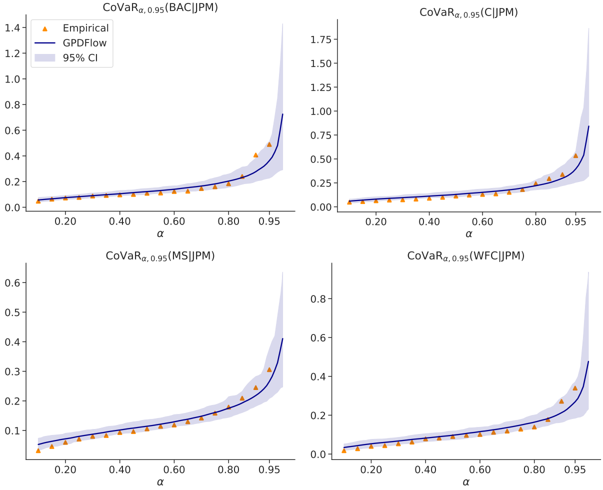


Application: Model Comparison

We compare GPDFlow with the best parametric mGPD (selected via AIC).



We condition on JPM having a negative log return above its 0.95 quantile.



Limitations

While GPDFlow provides a flexible and powerful framework for modelling multivariate threshold exceedances, it is not without limitations:

- **Assumption of asymptotic dependence**: GPDFlow is based on max-domain attraction, which may not be suitable for asymptotically independent data.
- **Lack of uncertainty quantification**: Estimating confidence intervals or posterior distributions for model parameters is challenging.
- **Training instability with disparate marginals**: When marginal scales differ substantially, a two-step estimation procedure is often required for stability.

Reference

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using real NVP. In *International Conference on Learning Representations*.

Kiriliouk, A., Rootzén, H., Segers, J., and Wadsworth, J. L. (2019). Peaks over thresholds modeling with multivariate generalized pareto distributions. *Technometrics*, 61(1):123–135.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64.

Rootzén, H., Segers, J., and Wadsworth, J. L. (2018). Multivariate generalized pareto distributions: Parametrizations, representations, and properties. *Journal of Multivariate Analysis*, 165:117–131.

Scan here for full manuscript if you are interested in our work!