Flexible Joint Modelling of Multivariate Extreme and Non-extreme Events

Chenglei Hu

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Mathematics & Statistics
College of Science and Engineering
University of Glasgow

Feb 2026

Abstract

In fields such as finance and environmental science, modelling the entire distribution of events with a particular focus on extremes is critical for risk management. Extreme Value Theory (EVT) offers a rigorous framework for such modelling. Initially developed to study the asymptotic behaviour of maxima of i.i.d. sequences, EVT was later extended to characterise the tails of distributions. A widely used result in univariate EVT is the peak-over-threshold (PoT) method, which approximates the tail of a distribution using the Generalised Pareto Distribution (GPD) above a sufficiently high threshold. This has motivated a "sliced" modelling framework that combines a separate distribution for the bulk (below threshold) with a GPD for the tail.

This thesis extends the sliced model to the multivariate setting and proposes three frameworks to either address the practical challenges arising in such extensions or provide alternate approaches for joint modelling of the bulk and tail.

Our first contribution is a multivariate analogue of the sliced model, combining a parametric bulk distribution with a multivariate GPD (mGPD) for the tail. The threshold separating bulk and tail is treated as a free parameter to avoid manual specification. Simulation studies demonstrate that the model robustly estimates marginal behaviour and both bulk and tail dependence, even under misspecification (e.g., when data are asymptotically independent but the model assumes asymptotic dependence). However, three limitations hinder scalability and realism in higher dimensions or large datasets. First, the mGPD is infinitely parameterised, with only a few closed-form representations available, risking bias if the true dependence deviates from these forms. Second, the piecewise construction introduces discontinuities at the bulk-tail boundary, both in the margins and in the dependence structure, which are unrealistic for large datasets. Third, structural inconsistency arises: while the mGPD is always asymptotically dependent, the bulk model (e.g., Gaussian) may be asymptotically independent, leading to conflicts in dependence representation. Moreover, the fixed dependence class of the mGPD limits its applicability in contexts such as spatial modelling, where tail dependence may vary with distance.

To address the first issue, we introduce GPDFlow, a novel mGPD framework in which the dependence structure is modelled using normalising flows, which is a flexible class of generative models. Unlike classic mGPDs, GPDFlow avoids closed-form constraints and instead learns a parameterised dependence structure through flows, with density evaluation performed numerically. GPDFlow explicitly transforms light-tailed distributions into heavy-tailed ones, overcoming typical limitations of generative models. It performs well in describing the data where only subsets of variables are extreme and outperforms standard mGPDs in estimating both marginal and tail dependence.

To address the issues of discontinuity and fixed asymptotic dependence, we develop a second framework combining the extended GPD (eGP) with a latent Gaussian model, implemented in the R-INLA package using the integrated nested Laplace approximation (INLA). The eGP is a sub-asymptotic distribution that retains the key properties of the GPD while avoiding the need for threshold specification, yielding a fully continuous model. Dependence is captured through latent Gaussian fields, ensuring coherence and continuity across the entire distribution. We illustrate this approach in a one-month-ahead spatio-temporal wildfire forecast application in Portugal, focusing on moderate and extreme burn areas. A two-stage ensemble design integrates environmental and historical data: in the first stage, an XGBoost model learns complex covariate patterns, producing pseudo-covariates that feed into the second-stage latent Gaussian model. This addresses key limitations of the INLA framework in handling high-dimensional covariates and obtaining future environmental inputs in retrospective analyses. The eGP model and associated priors are now fully implemented in R-INLA and publicly available to users.

Finally, to explore a purely deep learning-based solution without asymptotic constraints or rigid latent structures, we propose a model tailored to the EVA Data Challenge 2025, which involves estimating the expected number of daily precipitation extremes across a 5×5 spatial grid over 165 years. We use a long short-term memory (LSTM) network to encode spatio-temporal patterns and condition a denoising diffusion probabilistic model (DDPM) on the resulting hidden states. The model operates on log-transformed, zero-adjusted precipitation data. For comparison, we also develop a sliced model with conditional independent margins, using a Weibull distribution for the bulk and a GPD for the tail. The diffusion-based model performs better for five out of six target quantities in the data challenge evaluated at lower thresholds, and accurately captures tail heaviness, as validated by marginal GEV shape parameter analysis on simulated and real data.

Contents

Abstract **i**

Acknowledgements **xi**

Declaration **xii**

1 Introduction **1**

1.1 Univariate extreme value theory . . . 1

1.2 Multivariate extreme value theory . . . 6

1.3 Jointly modelling the bulk and tail . . . 22

1.4 Outline of the thesis . . . 25

2 Methodological framework **27**

2.1 Methods for Bayesian inference . . . 27

2.1.1 Metropolis–Hastings algorithm . . . 27

2.1.2 Gibbs sampling . . . 29

2.1.3 Slice sampling . . . 31

2.1.4 Automated factor slice sampling . . . 32

2.1.5 Integrated nested Laplace approximation . . . 34

2.2 XGBoost . . . 35

2.3 Deep learning . . . 38

2.3.1 Multilayer perceptron . . . 38

2.3.2 Long short-term memory . . . 39

2.3.3 Normalising flows . . . 41

2.3.4 Diffusion models . . . 44

3 A Bayesian multivariate extreme value mixture model 49

3.1 Introduction . . . 49

3.2 Multivariate extreme mixture model . . . 51

3.2.1 Multivariate generalised Pareto distribution . . . 51

3.2.2 Multivariate bulk distribution . . . 53

3.2.3 Multivariate extreme mixture distribution . . . 54

3.3 Bayesian inference . . . 56

3.3.1 Priors for the parameters in the bulk . . . 56

3.3.2 Prior for the threshold . . . 57

3.3.3 Priors for the parameters in the tail . . . 58

3.3.4 Posterior inference . . . 58

3.4 Simulations . . . 60

3.4.1 Well-specified scenarios (Scenario 1) . . . 60

3.4.2 Misspecification scenario (Scenario 2) . . . 62

3.5 Application . . . 63

3.5.1 Pre-processing and model specification . . . 65

3.5.2 Results . . . 66

3.6 Discussion . . . 71

3.6.1 Other distributions for the bulk and alternative representations for the tail 71

3.6.2 Model selection . . . 72

3.6.3 High-dimensional generalisation . . . 72

3.7 Conclusion . . . 73

3.8 Supplementary materials . . . 74

3.8.1 Code . . . 74

3.8.2 Proofs . . . 74

3.8.3 Stationarity check of the temperature data . . . 76

3.8.4 MCMC convergence diagnostics . . . 77

4 Multivariate GPD with normalising flows 79

4.1 Introduction . . . 79

4.2 Method . . . 81

4.2.1 Multivariate GPD . . . 81

4.2.2 Normalising flows . . . 84

4.2.3 GPDFlow 87

4.3 Inference 88

4.3.1 Likelihood and model estimation 88

4.3.2 Threshold selection 90

4.3.3 Hyperparameter tuning 92

4.4 Simulation 93

4.5 Application 97

4.6 Discussion 101

4.7 Supplementary material 103

4.7.1 Code and data 103

4.7.2 Proofs 103

4.7.3 Hyperparameter tuning results 104

4.7.4 Supporting plots 109

5 Moderate and extreme wildfire forecast **110**

5.1 Introduction 110

5.2 Data preparation 114

5.2.1 Wildfire data scope 114

5.2.2 Environmental covariates 115

5.3 Methods 116

5.3.1 Overview 116

5.3.2 Stage I: XGBoost 118

5.3.3 Stage II: Bayesian latent Gaussian modelling 120

5.4 Results 129

5.4.1 Model comparison 129

5.4.2 Posterior predictions 131

5.4.3 Covariates and latent effects 133

5.4.4 Posterior Distributions of eGP Parameters 136

5.5 Discussion 137

5.5.1 Data transformation and ξ in eGP 137

5.5.2 Similar performance to Gamma and Weibull likelihoods 138

5.5.3 Longer forecasting horizons 139

5.5.4 Choices of the model in stage 1 140

5.6 Conclusion 141
5.7 Supplementary material 142
5.7.1 Code and data 142
5.7.2 Derivation of the PC prior for $\kappa > 0$ 142
5.7.3 Covariates in XGBoost 144
5.7.4 Additional diagnostic plots 146

6 Modelling bulk and tail distributions with deep learning 148
6.1 Introduction 148
6.2 Data description and preliminary exploration 150
6.3 Methods 151
6.3.1 Long short-term memory 151
6.3.2 Diffusion models 152
6.3.3 Data transformation 153
6.3.4 Latent state modelling 154
6.3.5 Other explorations 155
6.4 Results 157
6.4.1 Evaluation metrics 157
6.4.2 Model tuning and comparison 158
6.4.3 Final submission results 160
6.5 Conclusion and discussion 162
6.5.1 Discussion 162
6.5.2 Limitation 163

7 Conclusion & future work 164
7.1 Future work 167

List of Tables

3.1 Tail shape parameters for the three well-specified simulation scenarios. Each scenario differs in the heaviness of the marginal tails, as determined by the shape parameters (γ_1, γ_2). 60

3.2 Average posterior means with 95% credible interval (CI) lengths in brackets, and coverage rates (CR) for all parameters across three well-specified scenarios. 61

3.3 Energy score (ES) and threshold-weighted energy score (twES) for the BEMM and Gaussian models. 71

3.1 Rank-normalised $\hat{R}$ and ESS (in parentheses) for the MCMC results in the data application. . . 77

4.1 Hyperparameter tuning results for GPDFlow in Simulation 1. 105

4.2 Hyperparameter tuning results for GPDFlow in Simulation 2. 107

4.3 Hyperparameter tuning results for GPDFlow in the data application. 108

5.1 Comparison of posterior predictive performance across model variants on the test set. 130

5.2 Comparison of prediction accuracy and uncertainty for model M1 across forecast horizons $h = 1, 2, 3$. 141

5.3 Environmental Covariates used in the XGBoost model. 145

5.4 Feature-engineered autoregressive covariates for fire count and burnt area. 146

6.1 Model comparison via the scaled squared error R_i between $\widetilde{P}_i(\boldsymbol{v}_i)$ and $\widehat{P}_i(\boldsymbol{v}_i)$; $R = \sum_{i=1}^{6} R_i$. 159

6.2 Final submission: 400 temporal block bootstrap samples ($b = 8000$). 162

List of Figures

1.1 Illustration of Block-Maxima method and Peak-over-Threshold method. 4

2.1 Illustrated architecture of an MLP with 3-dimensional input and 2-dimensional output. 38

2.2 Architecture of an LSTM block that updates the cell state $\boldsymbol{c}_{t-1}$ and hidden state $\boldsymbol{h}_{t-1}$ using the current input $\boldsymbol{x}_t$. 42

2.3 Forward (left) and backward (right) propagation in a layer of a Real NVP model. 44

3.1 Two-dimensional density contour plots derived from (3.11). 55

3.2 Boxplots comparing extremal dependence metrics ($\chi(r)$, $\bar{\chi}(r)$), and Kendall's τ for the bivariate extreme mixture model (BEMM, blue) and the bivariate mixture copula model (BMCM, orange) under Scenario 2 (Gaussian data, asymptotic independence). 64

3.3 Scatterplot of the negative residuals from model (3.19). 66

3.4 Histograms of the posterior distributions for all model parameters. 67

3.5 Joint posterior density of the thresholds and the binned mean log-likelihood surface. 68

3.6 Q–Q plots of posterior predictive quantiles (blue) from 3,000 replicated datasets against the empirical quantiles. 68

3.7 Based on 3,000 replicates from the posterior predictive distribution, the upper two plots present empirical values vs. means of the posterior prediction, each accompanied by a 95% credible band, of $\chi(r)$ and $\bar{\chi}(r)$. 69

3.8 Autocorrelation plots of the two sites after removing the seasonality and autocorrelation by (3.19). The red dashed line indicates the confidence band at 95% confidence level. 77

3.1 Trace plots for the MCMC results in the data application, with parameters ordered as in Figure 3.4. 78

4.1 Comparison between 1000 samples (left) from $f_{\boldsymbol{T}}(t_1,t_2) = \exp\{(t_1+1)/2+(t_2+2)/0.5\}t_1 < -1, t_2 < -2$, and samples (right) from estimated $\widehat{f}_{\boldsymbol{T}}$ from the mGPD data generated by $f_{\boldsymbol{T}}$, illustrating the unidentifiability of $f_{\boldsymbol{T}}$. 89

4.2 Boxplots of estimated pairwise χ (Top) and marginal parameters $\boldsymbol{\sigma}$, $\boldsymbol{\gamma}$ (Bottom two) across dimension $d = 2$, 3, and 5. 94

4.3 Theoretical and estimated density contour plots of the threshold exceedance data (top panel) and plot of $\log \mathbb{P}(Y_1 < q_{1,\alpha}, Y_2 > q_{2,0.99})$ (bottom). 96

4.4 Scatter plots of the negative log-return (left) of the five US banks and empirical $\chi_{1:d}(q)$ and $\omega_{1:d}(q)$ of the negative log-return calculated by Algorithm 6. 97

4.5 Plots of marginal densities of the threshold exceedance data (top panel) and tail dependence measures (bottom panel). 99

4.6 $\text{CoVaR}_{\alpha,\beta}(\cdot|\text{JPM})$ for the four banks, conditioned on JPM being in distress ($\beta = 0.95$). 101

4.7 Learning curves for the models in Simulation 1, plotted over epochs prior to early stopping. . . 106

4.8 Learning curves for the models in Simulation 2, plotted over epochs prior to early stopping. . . 107

4.9 Learning curves for the models in the data application, plotted over epochs prior to early stopping. 108

4.10 Marginal density estimation comparisons for bank BAC, C and WFC. 109

5.1 Histograms of fire count and burnt area at the council-month level, highlighting the prevalence of zeros. 115

5.2 Top: Average council-level fire count and burnt area. Bottom: Monthly average fire count and burnt area across all councils. 116

5.3 Diagram of the proposed two-stage wildfire forecasting framework combining XGBoost and INLA.117

5.4 Comparison of naive and window-based modelling configurations. 120

5.5 Time-series–aware cross-validation is implemented using yearly blocks for both hyperparameter tuning and the generation of one-month-ahead forecasts. 121

5.6 PC priors for κ based on exact KLD in (5.10) (left) and the approximated KLD around $\kappa = 1$ (5.11) (right) under penalisation rates $\lambda \in \{0.1, 0.5, 1, 2, 5, 10\}$. 128

5.7 Posterior predictive checks of threshold exceedance probabilities for fire count and burnt area in the test set (top two panels), and posterior predictive distributions of total fire count and burnt area in Portugal for 2017 (training set) and 2023 (test set) (bottom two panels). 132

5.8 SHAP values for the top 10 covariates in the XGBoost models for fire count and burnt area. . . 134

5.9 Posterior estimation of year effect $T(\cdot)$ (top) and XGBoost prediction effects $R(\cdot)$ in η^C (bottom left) and η^B (bottom right). 135

5.10 Posterior mean of the council-level spatio-temporal effect $G_c(\cdot)$ by grouped month (left), and average of the posterior mean of the district-level spatio-temporal effect $G_d(\cdot)$ (right), aggregated over all time indices. 136

5.11 Prior and posterior distributions of the hyperparameters ξ (left) and κ (right) in the eGP likelihood. 137

5.12 Densities of Gamma (shape = 1.80), Weibull (shape = 1.35) and eGP ($\kappa = 4.64$, $\xi = 0.45$) likelihoods, based on estimated hyperparameters. 139

5.13 Autocorrelation Function plots of average fire count and burnt area at council-monthly level. . 146

5.14 Posterior uncertainty of the spatio-temporal effects. 147

6.1 Tail dependence metrics $\chi(q)$ and $\omega(q)$ for run1, evaluated over $q \in (0.5, 0.99)$. 152

6.2 Architecture of the latent state framework for generating a new observation $\boldsymbol{y}_t$. 155

6.3 Predicted $\widetilde{P}_i(\boldsymbol{v}_i)$ from M_0 (mean over 1,000 simulations; bands are 2.5%–97.5% simulation quantiles) versus empirical $\widehat{P}_i(\boldsymbol{v}_i)$ aggregated over runs 1–4, for $i = 1, \ldots, 6$. 159

6.4 Site-wise ξ_{GEV} from annual maxima: run1 MLEs with asymptotic 95% CIs, and M_0 summaries from 1,000 simulated runs (mean with 2.5%–97.5% quantiles). 160

6.5 Site-wise Q–Q plots: M_0 quantiles versus empirical quantiles from run1; shaded regions are 95% simulation intervals from 1,000 simulated runs. 161

Acknowledgements

First and foremost, I want to say a heartfelt thank you to my supervisor, Dr Daniela Castro-Camilo. She's been so much more than a supervisor to me, more like a guiding light throughout this whole journey. Her passion for research, her joy for life, and the kind, thoughtful way she treats people have deeply influenced me, often in ways I only realised later. No matter what I brought to her, whether it worked out or not, she always found something positive in it. She had this incredible ability to spot the good in what I thought were trivial or failed attempts, and she offered new ways of thinking that helped me keep going when things got tough. Her encouragement, patience, and quiet confidence in me made all the difference. I honestly couldn't have made it through this PhD without her support.

I'm also really grateful to the College of Science & Engineering at the University of Glasgow for the scholarship that made this whole experience possible, and for giving me the chance to spend an extra four years in this amazing city.

A big thanks to my collaborators, Dr Ben Swallow, Dr Regina Baltazar Bispo, Prof Håvard Rue, and Dr Carlos C. DaCamara, for your time, ideas, and collaboration throughout my projects. I especially want to thank Regina for giving me the chance to turn one of my ideas into something real and useful. It meant a lot to know that my work could make a difference beyond academia.

I also want to thank Yan Gong. Her talk at the STOR-i Extremes Workshop in Lancaster opened my eyes to a new world of research—generative models and normalising flows. That moment of inspiration pushed me to explore a completely new direction in the final part of my PhD, and I'm so glad I did.

A special shout-out to Hiroyuki Sawano, Shoji Meguro, and Yu-Peng Chen for creating music that helped me survive the hard days. Their work kept me company during some really painful times, and I'll always be grateful for that comfort.

And finally, the biggest thank you goes to my parents. Their unconditional support, love, and trust in every choice I've made have been the foundation of everything. This PhD is just as much theirs as it is mine.

Declaration

I declare that all the work presented in this thesis, entitled "Flexible Joint modelling of Multivariate Extreme and Non-extreme Events" is the result of my own independent research, conducted under the supervision of Dr Daniela Castro-Camilo in the School of Mathematics and Statistics, University of Glasgow. I confirm that

- This thesis has not been submitted in whole or in part for any other degree or qualification at this or any other university or institution.
- All statistical analyses, models, and results presented herein were conducted and interpreted by me, unless otherwise stated.
- Any collaborative work, including co-authored publications, is clearly acknowledged

Chapters 3 to 6, excluding my supervisor's contributions, involve collaborative work. Authorship, presentations, and submission details are summarised below.

- Chapter 3
  - Co-author: Dr Ben Swallow (University of St Andrews)
  - Presentation (Contributed Talks):
    - STOR-i Extremes Workshop, Lancaster, United Kingdom, 2023
    - 6th International Conference on Advances in Extreme Value Analysis and Application to Natural Hazards, Venice, Italy, 2024
  - Submission: Submitted to *Journal of Computational Statistics & Data Analysis*
- Chapter 4
  - Co-authors: None
  - Presentations (Contributed Talks):

        - Royal Statistical Society (RSS) International Conference 2025, Edinburgh, United Kingdom, 2025
        - Generative AI Modelling for Extreme Events, Edinburgh, United Kingdom, 2025
        - The Glasgow–Edinburgh Extremes Network (online), 2025
    - Submission: Submitted to *Journal of Computational and Graphical Statistics*

- Chapter 5
    - Co-authors:
        - Dr Regina Baltazar Bispo (University of St Andrews/ NOVA School of Science and Technology)
        - Prof Håvard Rue (King Abdullah University of Science and Technology)
        - Dr Carlos C. DaCamara (University of Lisbon)
        - Dr Ben Swallow (University of St Andrews)
    - Presentation (Poster)
        - INLA: past, present, and future, Glasgow, United Kingdom, 2025
    - Submission: Submitted to *Environmetrics*.

- Chapter 6
    - Co-authors: None
    - Presentation: None
    - Submission: Will be submitted to a special issue of the *Extremes* journal.

Chapter 1

Introduction

Extreme events, defined as rare observations with exceptionally large or small values, arise frequently in fields such as engineering, finance, and environmental science. Examples include dam overtopping caused by probable maximum floods, sudden stock market crashes, and record-breaking heatwaves. Although rare, these events often have severe consequences. For instance, in July 2011, intense rainfall from Tropical Storm Nock-ten triggered devastating floods across 65 provinces in Thailand, resulting in more than 800 fatalities and an estimated USD 46 billion in economic losses.

Given the high impact and economic cost of such extremes, it is crucial to develop statistical tools capable of describing and quantifying them. In practice, interest often lies not in a single observed extreme, but rather in estimating the probability of events at least as extreme as the one observed. In other words, reliable extrapolation to unobserved data is essential. From a statistical perspective, this corresponds to studying the tail region of a distribution F that models the underlying events. Although F is typically unknown, its asymptotic tail behaviour can be well approximated using methods from Extreme Value Theory (EVT).

1.1 Univariate extreme value theory

EVT was first formally established by Fisher and Tippett (1928), who studied the limiting distribution of the maximum

$$M_n = \max(Y_1, \dots, Y_n),$$

where $Y_1, \ldots, Y_n$ are independent and identically distributed (i.i.d.) random variables with cumulative distribution function (CDF) F. The distribution of M_n can be written as

$$\begin{aligned}\mathbb{P}(M_n < y) &= \mathbb{P}(Y_1 < y, \ldots, Y_n < y)\\ &= \{\mathbb{P}(Y_1 < y)\}^n\\ &= \{F(y)\}^n,\end{aligned}$$

which is not particularly useful, as it depends on the unknown F. However, if we can find a sequence of constants $\alpha_n > 0$ and β_n, $n = 1, 2, \ldots$ such that the normalised maximum

$$\frac{M_n - \beta_n}{\alpha_n}$$

converge in distribution to a nondegenerate limiting distribuiton $G(x)$ as $n \to \infty$, i.e.,

$$\lim_{n\to\infty} F^n(\alpha_n x + \beta_n) = G(x), \tag{1.1}$$

then the following Fisher–Tippett–Gnedenko Theorem (De Haan and Ferreira, 2006, p. 6) provides a unified characterisation of G, analogous to the Gaussian limit in the classical central limit theorem.

Theorem 1 (Fisher–Tippett–Gnedenko Theorem)**.** *The nondegenerate distribution $G(x)$ in* (1.1) *is in the form*

$$G(x;\gamma) = \exp\left(-(1+\gamma x)^{-1/\gamma}\right), \quad 1 + \gamma x > 0.$$

When $\gamma = 0$, $G(x;0) = \lim_{\gamma\to 0} G(x;\gamma) = \exp(-\exp(-x))$. This family, known as the generalised extreme value (GEV) distribution, encompasses the Fréchet ($\gamma > 0$), Gumbel ($\gamma = 0$), and Weibull ($\gamma < 0$) distributions.

A crucial condition of the theorem is the existence of a non-degenerate limit G. If a distribution F satisfies this property, it is said to belong to the max-domain of attraction of G. A detailed account of the necessary and sufficient conditions for this, as well as explicit forms of α_n and β_n can be found in De Haan and Ferreira (2006). Fortunately, most commonly used distributions fall within a max-domain of attraction. Below is an example of using Theorem 1 to show that the standard Fréchet distribution is in the max-domain of attraction of itself.

Example 1.1.1. *Consider the standard Fréchet distribution with*

$$F(x) = \exp\{-1/x\}, \quad x > 0.$$

Setting $\alpha_n = n$ and $\beta_n = 0$, we obtain

$$\lim_{n\to\infty} F^n(nx) = \lim_{n\to\infty} \exp\left\{n \cdot \left(-\frac{1}{nx}\right)\right\} = F(x).$$

In practice, when modelling M_n with a GEV distribution, the normalising constants α_n and β_n need not be explicitly known. Since they are fixed given n, they can be absorbed into the location and scale parameters in the GEV distribution:

$$\begin{aligned}\mathbb{P}\left(M_n \le x\right) &= \mathbb{P}\left(\frac{M_n - \beta_n}{\alpha_n} \le \frac{x - \beta_n}{\alpha_n}\right)\\ &\approx G\left(\frac{x-\beta_n}{\alpha_n}; \gamma\right)\\ &= \exp\left(-\left[1 + \gamma\left(\frac{x-\beta_n}{\alpha_n}\right)\right]_+^{-1/\gamma}\right)\\ &= \exp\left(-\left[1 + \gamma\left(\frac{x-\mu}{\alpha}\right)\right]_+^{-1/\gamma}\right)\\ &:= G(x; \mu, \alpha, \gamma),\end{aligned} \tag{1.2}$$

where $\mu = \beta_n$ is the location parameter, $\alpha = \alpha_n$ is the scale parameter, and γ is the shape parameter. Thus, the block maxima of i.i.d. sequences can be modelled using the three-parameter GEV distribution, laying the foundation for the block maxima method. Before introducing this method, we present a non-stationary version of the Fisher–Tippett–Gnedenko theorem, starting with the $D(u_n)$ condition.

Definition 1. *For random vector $(Y_1, \dots, Y_n)$ and any integer indices $1 < i_1 < \dots < i_p < j_1 < \dots < j_q < n$ with $j_1 - i_p > l_n$ and $l_n = o(n)$, the $D(u_n)$ condition is said to be held if*

$$\begin{aligned}&|\mathbb{P}(Y_{i_1} < u_n, \dots Y_{i_p} < u_n, Y_{j_1} < u_n, \dots Y_{j_q} < u_n)\\ &\quad - \mathbb{P}(Y_{i_1} < u_n, \dots Y_{i_p} < u_n)\mathbb{P}(Y_{j_1} < u_n, \dots Y_{j_q} < u_n)| \le \epsilon(n, l_n)\end{aligned}$$

where $\epsilon(n, l_n) \to 0$ as $n \to \infty$.

This condition generalises independence: while completely independent sequences satisfy $\epsilon(n, l_n) = 0$, $D(u_n)$ allows short-range dependence, provided that sufficiently distant observations behave approximately independently at extreme levels. For the random vector that satisfies the $D(u_n)$ condition, the following theorem demonstrates that its maxima can still be approximated by a GEV distribution.

Theorem 2 (Leadbetter (1983))**.** *Let $Y_1, \ldots, Y_n$ be a random vector with maximum $M_n = \max(Y_1, \ldots, Y_n)$. Suppose there exists $\alpha_n > 0$ and β_n, and an nondegenerate distribuiton G such that the $D_n(u_n)$ condition holds for $u_n = \alpha_n x + \beta_n,\ x \in \mathbb{R}$, and*

$$\mathbb{P}\left(\frac{M_n - \beta_n}{\alpha_n} \leq x\right) \to G(x).$$

Then G is a generalised extreme value distribution.

With the above preparation, the block maxima method can be naturally introduced. The idea is straightforward: divide observations into blocks (commonly by year) and model the maximum of each block using the GEV distribution. In environmental applications, such as precipitation or temperature analysis, the $D(u_n)$ condition is typically satisfied for daily data when grouped annually, thus well justifying the use of GEV models. The left panel of Figure 1.1 illustrates the block maxima approach.

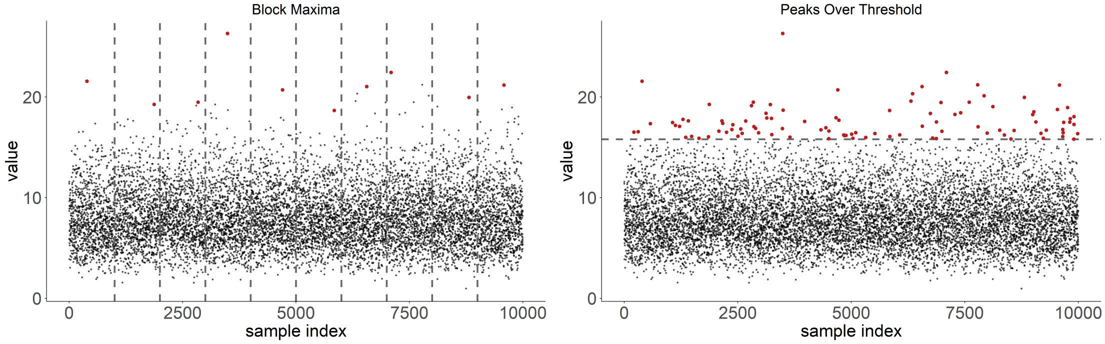

Figure 1.1: Illustrations of the block maxima method (left) and the peak-over-threshold method (right). In each plot, 10,000 i.i.d. samples are generated from a Gamma distribution with scale parameter 1 and shape parameter 8. The red dots denote the observations used in the block maxima or peak-over-threshold approaches. In the left panel, the grey vertical lines partition the samples into 10 blocks. In the right panel, the grey horizontal line marks the threshold, corresponding to the 0.99-quantile of the Gamma distribution.

One limitation of the block maxima method is that it relies solely on the maximum observation within each block and discards the near-maximum values that also contain valuable information about the tail behaviour of the underlying distribution. To address this inefficiency, we return to the key relation in equation (1.1), which underpins the Fisher–Tippett–Gnedenko theorem. By taking logarithms on both sides of (1.1) and applying a first-order Taylor expansion, we obtain

$$1 - F(x) \approx -\frac{1}{n} \log G\left(\frac{x-\beta_n}{\alpha_n}\right) = -\frac{1}{n} \log G(x; \mu, \alpha, \gamma) \tag{1.3}$$

This suggests that, instead of fitting a GEV distribution to block maxima, we can directly approximate the upper tail of F using a GEV distribution. To eliminate the normasling factor $1/n$ in (1.3), we introduce a fixed high threshold u such that

$$1 - F(u) \approx -\frac{1}{n} \log G(u; \mu, \alpha, \gamma).$$

Taking the ratio between (1.3) and $1 - F(u)$ yields a conditional exceedance probability, leading to the following result.

Theorem 3 (Pickands–Balkema–De Haan theorem)**.** *Let Y be a random variable with CDF F in the max-domain of attraction of a GEV distribution $G(\cdot; \mu, \alpha, \gamma)$. Then, for a sufficiently large threshold u,*

$$\mathbb{P}(Y - u < x \mid Y > u) \to H(x),$$

where

$$H(x) = 1 - \left(1 + \frac{\gamma x}{\sigma_u}\right)_+^{-1/\gamma}, \quad \sigma_u = \alpha + \gamma(u - \mu).$$

Here, the sign $(\cdot)_+$ means that the term inside the parentheses must be nonnegative. The distribution $H(x)$ is a generalised Pareto distribution (GPD), with shape parameter γ and scale parameter σ_u. The shape parameter, identical to that of the corresponding GEV distribution, governs the tail behaviour and the support of the GPD:

1. $\gamma > 0$: GPD is supported on $[0, \infty)$ and exhibits a Pareto-type (heavy) tail.

2. $\gamma = 0$: GPD is supported on $[0, \infty)$ and reduces to the exponential distribution, corresponding to a light tail.

3. $\gamma < 0$: GPD has finite support on $[0, -\sigma/\gamma]$.

The scale parameter σ_u depends on the GEV parameters (μ, α, γ) and the threshold u, and is therefore threshold-specific. In practice, the threshold is fixed to ensure the identifiability of σ_u.

For distributions in the max-domain of attraction of a GEV, Theorem 3 implies that the tail beyond a sufficiently high threshold can be approximated by a GPD:

$$1 - F(y) \approx \{1 - F(u)\}H(y - u), \quad y > u$$

This motivates the peak-over-threshold (POT) method for modelling extremes: select a high threshold, and fit a GPD to the exceedances. The right panel of Figure 1.1 illustrates this approach.

A critical issue in applying the POT method is threshold selection, which entails a bias-variance trade-off. If the threshold is too low, the asymptotic justification for the GPD approximation may fail, introducing bias. Conversely, if the threshold is too high, the number of exceedances becomes too small for reliable estimation, inflating variance. A common strategy for threshold choice is to use diagnostic plots. The idea is to identify quantities with theoretically stable behaviour under the GPD model, and to examine their empirical behaviour across a range of thresholds. For example, for a random variable Y with CDF F in the max-domain of attraction of a GEV distribution $G(\cdot;\mu,\alpha,\gamma)$, Theorem 3 indicates that for a sequences of sufficiently high threshold $u_k,\ k = 1, 2, \ldots$,

$$\mathbb{P}(Y - u_k < x \mid Y > u_k) \to 1 - \left(1 + \frac{\gamma x}{\sigma_{u_k}}\right)_+^{-1/\gamma},$$

with $\sigma_{u_k} = \alpha + \gamma(u_k - \mu)$. This implies that both the shape parameter γ and the modified scale parameter $\tilde{\sigma} = \sigma_{u_k} - \gamma u_k$ remain constant across valid thresholds. Hence, by plotting the estimated γ and $\tilde{\sigma}$ against candidate thresholds, one may identify the smallest threshold at which both estimates stabilise, and adopt this as the working threshold.

1.2 Multivariate extreme value theory

When the joint tail behaviour of multiple random variables is of interest, it is natural to extend the univariate EVT framework, which is based on the GEV distribution and GPD, to multivariate settings. We begin with the multivariate analogue of the max-domain of attraction condition in (1.1). Throughout, we use $\vee$ to denote the element-wise maximum: for vectors $\boldsymbol{a} = (a_1, \ldots, a_d)$ and $\boldsymbol{b} = (b_1, \ldots, b_d)$, $\boldsymbol{a} \vee \boldsymbol{b} = (\max(a_1, b_1), \ldots, \max(a_d, b_d))$. Similarly, $\wedge$ denotes the element-wise minimum. Let $\boldsymbol{Y}_i,\ i = 1, \ldots, n$ be i.i.d. d-dimensional vectors with CDF F. We say F lies in the max-domain of attraction of a nondegenerating distribution G if there exists normalising sequences $\boldsymbol{\alpha}_n > \mathbf{0}$ and $\boldsymbol{\beta}_n$ such that

$$\lim_{n\to\infty} F^n(\boldsymbol{\alpha}_n \boldsymbol{x} + \boldsymbol{\beta}_n) = G(\boldsymbol{x}). \tag{1.4}$$

Intuitively, this states that, with appropriate normalising, the distribution of the componentwise maximum $\boldsymbol{M}_n = \bigvee_{i=1}^n \boldsymbol{Y}_i$ converges to a nondegenerating distribution G:

$$\mathbb{P}\left(\frac{\boldsymbol{M}_n - \boldsymbol{\beta}_n}{\boldsymbol{\alpha}_n} \le \boldsymbol{x}\right) = \left\{\mathbb{P}\left(\frac{\boldsymbol{X}_1 - \boldsymbol{\beta}_n}{\boldsymbol{\alpha}_n} \le \boldsymbol{x}\right)\right\}^n = F^n(\boldsymbol{\alpha}_n \boldsymbol{x} + \boldsymbol{\beta}) \to G(\boldsymbol{x})$$

Marginally, (1.4) reduces to the univariate condition in (1.1):

$$\lim_{n\to\infty} F_j^n(\alpha_{n,j} x_j + \beta_{n,j}) = G_j(x_j),$$

where F_j and G_j are the marginal distribution of F and G, respectively. Thus, if such a limit G exists, each of its marginals must be GEV distributions. Accordingly, G in (1.4) is called a multivariate generalised extreme value (mGEV) distribution (Beirlant et al., 2006). Without loss of generality, we can standardise G to standard Fréchet margins via the componentwise transformation

$$T(\boldsymbol{x}) = (T_1(x_1), \ldots, T_d(x_d)) = \left(-\frac{1}{\log G_1(x_1)}, \ldots, -\frac{1}{\log G_d(x_d)}\right) \tag{1.5}$$

and then only focus on its dependence structure.

Let G_* denote the standardised G. Then

$$G(\boldsymbol{x}) = G_*(T(\boldsymbol{x})) = G_*\left(-\frac{1}{\log G_1(x_1)}, \ldots, -\frac{1}{\log G_d(x_d)}\right),$$

and conversely,

$$G_*(\boldsymbol{z}) = G(T^{-1}(\boldsymbol{z})) = G\{G_1^{\leftarrow}(\exp\{-1/z_1\}), \ldots, G_d^{\leftarrow}(\exp\{-1/z_d\})\},$$

where $G_j^{\leftarrow}$ is the quantile function of the univariate GEV distribution G_j, given by

$$G_j^{\leftarrow}(p) = \begin{cases} \mu_j + \frac{\alpha_j}{\gamma_j}[(-\log p)^{-\gamma_j} - 1] & p \in (0,1), \gamma_j \neq 0 \\ \mu_j - \alpha_j \log(-\log p) & p \in (0,1), \gamma_j = 0 \end{cases}$$

with location μ_j, scale α_j, and shape γ_j.

To facilitate characterising G and G_*, we first introduce the following concept.

Definition 2. *A distribution P is said to be max-stable if there exists $\boldsymbol{a}_n > \mathbf{0}$ and $\boldsymbol{b}_n$ such that*

$$P^n(\boldsymbol{a}_n\boldsymbol{x} + \boldsymbol{b}_n) = P(\boldsymbol{x})$$

The mGEV distribution G is max-stable. To see this, note that for any $t > 0$,

$$\begin{aligned}
&F^n(\boldsymbol{\alpha}_n\boldsymbol{x} + \boldsymbol{\beta}_n) \to G(\boldsymbol{x}),\\
&F^{[nt]}(\boldsymbol{\alpha}_{[nt]}\boldsymbol{x} + \boldsymbol{\beta}_{[nt]}) \to G(\boldsymbol{x}),\\
&F^{[nt]}(\boldsymbol{\alpha}_n\boldsymbol{x} + \boldsymbol{\beta}_n) = (F^n(\boldsymbol{\alpha}_n\boldsymbol{x} + \boldsymbol{\beta}_n))^{[nt]/n} \to G^t(\boldsymbol{x}).
\end{aligned}$$

By Theorem 14.2 in Billingsley (1995), there exist functions $\boldsymbol{a}(t)$ and $\boldsymbol{b}(t)$ such that $\boldsymbol{\alpha}_{[nt]}/\boldsymbol{\alpha}_n \to \boldsymbol{a}(t)$, $(\boldsymbol{\beta}_{[nt]} - \boldsymbol{\beta}_n)/\boldsymbol{\alpha}_{\boldsymbol{n}} \to \boldsymbol{b}(t)$, and

$$G^t(\boldsymbol{a}(t)\boldsymbol{x} + \boldsymbol{b}(t)) = G(\boldsymbol{x}).$$

This shows raising G to a power corresponds only to a location–scale transformation; the resulting distribution is still a multivariate GEV with the same dependence structure (that is, characterised by the same G_*).

We can also show that the standardised distribution G_* is also max-stable. Since the marginals of a max-stable distribution are themselves max-stable, the transformation T satisfies

$$\begin{aligned}
T(\boldsymbol{a}(t)\boldsymbol{x} + \boldsymbol{b}(t)) &= \left(-\frac{1}{\log G_1(a_1(t)x_1 + b_1(t))}, \dots, -\frac{1}{\log G_d(a_d(t)x_d + b_d(t))}\right)\\
&= \left(-\frac{t}{\log G_1(x_1)}, \dots, -\frac{t}{\log G_d(x_d)}\right)\\
&= tT(\boldsymbol{x}),
\end{aligned}$$

with inverse transformation

$$\boldsymbol{a}(t)T^{-1}(\boldsymbol{y}) + \boldsymbol{b}(t) = T^{-1}(t\boldsymbol{y}).$$

Therefore,

$$\begin{aligned} G_*^t(t\boldsymbol{z}) &= G^t(T^{-1}(t\boldsymbol{z})) \\ &= G^t(\boldsymbol{a}(t)T^{-1}(\boldsymbol{z}) + \boldsymbol{b}(t)) \\ &= G(T^{-1}(\boldsymbol{z})) \\ &= G_*(\boldsymbol{z}). \end{aligned}$$

With normalising constants $\boldsymbol{c}(t) = t$ and $\boldsymbol{d}(t) = 0$, this can be expressed as

$$G_*^t(\boldsymbol{c}(t)\boldsymbol{z} + \boldsymbol{d}(t)) = G_*(\boldsymbol{z}). \tag{1.6}$$

Definition 3. *A distribution P is called max-infinitely divisible (max-id) if for every integer k, $P^{1/k}$ is still a distribution.*

Clearly, every max-stable distribution is max-id. The following Proposition shows that any max-id distribution admits a representation in terms of a Radon measure, also known as the *exponent measure*.

Proposition 4 (Resnick (1987), Chapter 5)**.** *If G is max-id, then for some $\boldsymbol{l} \in [-\infty, \infty)^d$, there exists an exponent measure μ on $E := [\boldsymbol{l}, \boldsymbol{\infty}]\backslash\{\boldsymbol{l}\}$ satisfying*

1. $\mu(E\backslash[-\infty, \infty)^d) = 0$
2. $x_j = l_j = -\infty$ *for some* $j = 1, 2, \ldots, d$ *implies* $\mu([-\boldsymbol{\infty}, \boldsymbol{x}]^c) = \infty$

such that

$$G(\boldsymbol{x}) = \begin{cases} \exp\{-\mu([-\boldsymbol{\infty}, \boldsymbol{x}]^c)\} & \boldsymbol{x} \geq \boldsymbol{l} \\ 0 & \textit{otherwise}. \end{cases}$$

Let μ_* denote the exponent measure associated with G_*. Since G_* has standard Fréchet margins with positive support, we can take $\boldsymbol{l} = \boldsymbol{0}$ so μ_* is concentrated on $E = [\boldsymbol{0}, \boldsymbol{\infty}]\backslash\{\boldsymbol{0}\}$, and

$$G_*(\boldsymbol{z}) = \exp\{-\mu_*([\boldsymbol{0}, \boldsymbol{z}]^c)\}, \quad \boldsymbol{z} \in E.$$

From (1.6), we know G_* is max-stable:

$$G_*^t(t\boldsymbol{z}) = G_*(\boldsymbol{z}), \quad t > 0$$

This implies the following homogeneity property for μ_*:

$$\mu_*(B) = t\mu_*(tB) \tag{1.7}$$

for a Borel set $B \subset E$ and $tB = \{tb : b \in B\}$. The homogeneity property suggests a natural separation of size and direction in μ_*: the size component is universal and less informative, while the directional component captures dependence. Now, fix a norm $\|\cdot\|$ and let $\Xi = \{\boldsymbol{z} \in \mathbb{R}^d : \|\boldsymbol{z}\| = 1\}$ denote the unit sphere, we can define a polar mapping $\mathcal{T} : E \to (0, \infty) \times \Xi$ by

$$\mathcal{T}(\boldsymbol{z}) = (\|\boldsymbol{z}\|, \|\boldsymbol{z}\|^{-1}\boldsymbol{z}).$$

A finite measure S, called the spectral measure, can then be defined on Ξ by

$$S(A) = \mu_*(\{\boldsymbol{z} : \|\boldsymbol{z}\| \geq 1, \boldsymbol{z}/\|\boldsymbol{z}\| \in A\})$$

for Borel subset $A \subset \Xi$. The homogeneity condition (1.7) translates into

$$\mu_*(\{\boldsymbol{z} \in E : \|\boldsymbol{z}\| \geq r, \boldsymbol{z}/\|\boldsymbol{z}\| \in A\}) = r^{-1}S(A), \tag{1.8}$$

showing that μ_* factors into a measure R in the radial measure $R(\{\boldsymbol{z} : \|\boldsymbol{z}\| > r\}) = r^{-1}$ and the spectral measure S. Equivalently, (1.8) can be written as a pushforward measure

$$\mu_* \circ \mathcal{T}^{-1}(\mathrm{d}r, \mathrm{d}\boldsymbol{\omega}) = r^{-2}\mathrm{d}rS(\mathrm{d}\boldsymbol{\omega}). \tag{1.9}$$

Using (1.9), the distribution of $G_*(\boldsymbol{z})$ can be formulated as

$$\begin{aligned}
-\log G_*(\boldsymbol{z}) &= \mu_*([\mathbf{0}, \boldsymbol{z}]^c) \\
&= \int\int_{\mathcal{T}([\mathbf{0},\boldsymbol{z}]^c)} r^{-2}\mathrm{d}rS(\mathrm{d}\boldsymbol{\omega}) \\
&= \int_{\Xi} S(\mathrm{d}\boldsymbol{\omega}) \left(\int_{[r > \min_{j=1}^{d}(z_j/\omega_j)]} r^{-2}\mathrm{d}r \right) \\
&= \int_{\Xi} \max_{j=1}^{d} \left(\frac{\omega_j}{z_j} \right) S(\mathrm{d}\boldsymbol{\omega}).
\end{aligned}$$

The standard Fréchet margins of G_* requires that

$$\frac{1}{z_j} = \int_\Xi \max\left(0, \dots, \frac{\omega_j}{z_j}, \dots, 0\right) S(\mathrm{d}\boldsymbol{\omega}) = \frac{1}{z_j}\int_\Xi \omega_j S(\mathrm{d}\boldsymbol{\omega}), \quad j = 1, \dots, d$$

and therefore

$$\int_\Xi \omega_j S(\mathrm{d}\boldsymbol{\omega}) = 1 \quad \text{for } j = 1, \dots, d$$

The above results are summarised in the following theorem.

Proposition 5 (Resnick (1987), Chapter 5)**.** *If G_* is a multivariate extreme value distribution with standard Fréchet margins, then G_* has the form*

$$G_*(\boldsymbol{z}) = \exp\{-V(\boldsymbol{z})\},$$

where

$$V(\boldsymbol{z}) = \int_\Xi \max_{j=1}^{d}\left(\frac{\omega_j}{z_j}\right) S(d\boldsymbol{\omega})$$

is called the exponent function, and S is a finite measure S on $\Xi = \{\boldsymbol{z} \in \mathbb{R}^d : \|\boldsymbol{z}\| = 1\}$ that satisfies

$$\int_\Xi \omega_j S(d\boldsymbol{\omega}) = 1 \quad \textit{for } j = 1, \dots, d$$

To obtain a parametric form of G_*, both the norm $\|\cdot\|$ and the spectral measure S (or equivalently, its normalised probability measure) must be specified. We demonstrate the choice of these two terms for deriving the logistic family of the mGEV distribution. Let the norm be ℓ^1 (sum): $\|\boldsymbol{x}\| = |x_1| + \cdots + |x_d|$. Under this choice, the support of the spectral measure is the simplex $\Xi = \{\boldsymbol{\omega} \in [\mathbf{0}, \infty) : \omega_1 + \cdots + \omega_d = 1\}$, since $\boldsymbol{z} \geq \mathbf{0}$ as a consequence of the standard Fréchet margins of G_*. The total mass of S on Ξ is

$$S(\Xi) = \int_\Xi 1 \cdot S(\mathrm{d}\boldsymbol{\omega}) = \int_\Xi (\omega_1 + \cdots + \omega_d) \cdot S(\mathrm{d}\boldsymbol{\omega}) = d.$$

Thus, S can be normalised to a probability measure Q via $Q = S/d$. When $d = 2$, taking the density of Q as

$$q(\omega) = \frac{1}{2}\left(\alpha^{-1} - 1\right)\{\omega(1-\omega)\}^{-1-1/\alpha}\left\{\omega^{-1/\alpha} + (1-\omega)^{-1/\alpha}\right\}^{\alpha-2}; \quad \omega \in (0, 1)$$

yields the logistic family of bivariate GEV distributions (Coles et al., 2001):

$$G_*(z_1, z_2) = \exp\left\{-\left(z_1^{-1/\alpha} + z_2^{-1/\alpha}\right)^{\alpha}\right\}, \quad z_1 > 0, z_2 > 0. \tag{1.10}$$

The dependence structure in (1.10) is controlled by $\alpha \in (0, 1)$. As $\alpha \to 0$, G_* converges to

$$G_*(z_1, z_2) \to \exp\{-\max(z_1^{-1}, z_2^{-1})\},$$

indicating that Z_1 and Z_2 become perfectly dependent. In contrast, as $\alpha \to 1$,

$$G_*(z_1, z_2) \to \exp\{-(z_1^{-1} + z_2^{-1})\},$$

and Z_1 and Z_2 approach independence.

In Proposition 5, the exponent function was defined in terms of a spectral measure. It can also be expressed through the exponent measure μ_*:

$$V(\boldsymbol{z}) = \mu_*([\boldsymbol{0}, \boldsymbol{\infty})\backslash[\boldsymbol{0}, \boldsymbol{z}]), \quad \boldsymbol{z} \in [\boldsymbol{0}, \boldsymbol{\infty}].$$

Evaluating V at $1/\boldsymbol{z}$ leads to another important tail dependence characterising function, namely stable tail dependence function (stdf, Beirlant (2004)) ℓ:

$$\begin{aligned} \ell(\boldsymbol{z}) &= V(1/z_1, \ldots, 1/z_d) \\ &= \int_{\Xi} \max_{j=1}^{d} (\omega_j z_j)\, S(\mathrm{d}\boldsymbol{\omega}), \quad \boldsymbol{z} \in [\boldsymbol{0}, \boldsymbol{\infty}]. \end{aligned} \tag{1.11}$$

The stdf ℓ can equivalently be defined in terms of G with arbitrary GEV margins:

$$\ell(\boldsymbol{z}) = -\log G(G_1^{\leftarrow}(\exp\{-z_1\}), \ldots, G_d^{\leftarrow}(\exp\{-z_d\})), \quad \boldsymbol{z} \in [\boldsymbol{0}, \boldsymbol{\infty}]$$

and conversely,

$$-\log G(\boldsymbol{x}) = \ell(-\log G_1(x_1), \ldots, -\log G_d(x_d)), \quad \boldsymbol{x} \in \{\boldsymbol{x} : G(\boldsymbol{x}) > 0\}. \tag{1.12}$$

Thus, although introduced via the exponent function of an mGEV distribution with Fréchet margins, ℓ more generally characterises the dependence structure of G, invariant under any monotone marginal transformations.

The purpose of introducing stdf is that, in practice, it is more convenient to deal with this function than the exponent or spectral measure. We will look into the stdf a bit more, as it will be frequently mentioned when we define a multivariate generalised Pareto distribution.

Since the marginal distributions do not affect ℓ, we study it under the standardised distribution G_*. Using the convexity of the max function, the homogeneity of μ_* and the standard Fréchet margins of G_*, the stdf ℓ has the following properties:

1. ℓ is convex;
2. $\ell(c\boldsymbol{z}) = c\ell(\boldsymbol{z})$ for $0 < c < \infty$;
3. $\ell(\boldsymbol{e}_j) = 1$ for unit vector $\boldsymbol{e}_j,\ j = 1, \dots, d$ in $\mathbb{R}^d$;

Another useful representation of ℓ is

$$\ell(\boldsymbol{z}) = \mathbb{E}[\max(\boldsymbol{z}\boldsymbol{V})], \quad z \in [\boldsymbol{0}, \boldsymbol{\infty}], \tag{1.13}$$

where $\boldsymbol{V}$ is a random variable with values in $[\boldsymbol{0}, \boldsymbol{\infty})$ with $\mathbb{E}(V_j) = 1$ (Rootzén et al., 2018b). This representation can be obtained by normalising the spectral measure S to a probability measure $Q = S/m$, where $m = S(\Xi)$, and set $\boldsymbol{V} = m\boldsymbol{W},\ \boldsymbol{W} \sim Q$. Then

$$\ell(\boldsymbol{z}) = \int_{\Xi} \max_{j=1}^{d} (\omega_j z_j)\, S(\mathrm{d}\boldsymbol{\omega}) = \int_{\Xi} \max_{j=1}^{d} (m w_j z_j)\, P(\mathrm{d}\boldsymbol{w}) = \mathbb{E}[\max(\boldsymbol{z}\boldsymbol{V})].$$

Conversely, if a function $\ell : [\boldsymbol{0}, \boldsymbol{\infty}] \to [0, \infty]$ can be expressed in the form (1.13), then it is a valid stdf Segers (2012). The corresponding spectral measure on Ξ defined as $S = gP_V \circ T_*^{-1}$ for a map $T_*(\boldsymbol{v}) = \boldsymbol{v}/\|\boldsymbol{v}\|$, weight function $g(\boldsymbol{v}) = \|\boldsymbol{v}\|$, and probability measure P_V of random variable V. Equivalently

$$S(A) = \int \mathbb{1}\left\{\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} \in A\right\} \|v\| P_V(\mathrm{d}\boldsymbol{v}) = \mathbb{E}\left(\mathbb{1}\left\{\frac{\boldsymbol{V}}{\|\boldsymbol{V}\|} \in A\right\} \|\boldsymbol{V}\|\right), \quad A \in \Xi.$$

Multivariate Threshold Exceedance Modelling

We can construct a multivariate generalised Pareto distribution (mGPD) in a manner analogous to the univariate case. To start with, suppose the max-stable domain of attraction condition in (1.4) holds for the CDF F of a random vector $\boldsymbol{Y}$. Then the survival probability of the normalised

$\boldsymbol{Y}$ has the limit

$$\mathbb{P}\left(\frac{\boldsymbol{Y}-\boldsymbol{\beta}_n}{\boldsymbol{\alpha}_n} \nleq \boldsymbol{x}\right) = 1 - F(\boldsymbol{\alpha}_n\boldsymbol{x}+\boldsymbol{\beta}_n) \to -\frac{1}{n}\log G(\boldsymbol{x}).$$

Without loss of generality, we can evaluate this probability at $\mathbf{0}$ and use the result as a conditioning term in a conditional distribution to remove the effect of n:

$$\mathbb{P}\left(\frac{\boldsymbol{Y}-\boldsymbol{\beta}_n}{\boldsymbol{\alpha}_n} \nleq \mathbf{0}\right) = 1 - F(\boldsymbol{\beta}_n) \to -\frac{1}{n}\log G(\mathbf{0}).$$

Thus, $\boldsymbol{\beta}_n$ can be interpreted as a threshold, and the set $\{\boldsymbol{Y} \nleq \boldsymbol{\beta}_n\}$ defines threshold exceedance events, i.e. at least one component exceeds the threshold. Now, conditioning on $\{\boldsymbol{Y} \nleq \boldsymbol{\beta}_n\}$, a straightforward calculation shows that the distribution of the conditional random vector $(\boldsymbol{Y} - \boldsymbol{\beta}_n)/\boldsymbol{\alpha}_n \mid \boldsymbol{Y} \nleq \boldsymbol{\beta}_n$ converges to

$$\begin{aligned}
&\mathbb{P}\left(\frac{\boldsymbol{Y}-\boldsymbol{\beta}_n}{\boldsymbol{\alpha}_n} < \boldsymbol{x} \;\middle|\; \boldsymbol{Y} \nleq \boldsymbol{\beta_n}\right)\\
=&\frac{F(\boldsymbol{\alpha}_n\boldsymbol{x}+\boldsymbol{\beta}_n) - F(\boldsymbol{\alpha}_n(\boldsymbol{x}\wedge\mathbf{0})+\boldsymbol{\beta}_n)}{1-F(\boldsymbol{\alpha}_n\boldsymbol{x}+\boldsymbol{\beta}_n)}\\
=&\frac{1-F(\boldsymbol{\alpha}_n(\boldsymbol{x}\wedge\mathbf{0})+\boldsymbol{\beta}_n) - [1-F(\boldsymbol{\alpha}_n\boldsymbol{x}+\boldsymbol{\beta}_n)]}{1-F(\boldsymbol{\alpha}_n\boldsymbol{x}+\boldsymbol{\beta}_n)}\\
\to&\frac{1}{\log G(\mathbf{0})}\log\frac{G(\boldsymbol{x}\wedge\mathbf{0})}{G(\boldsymbol{x})} := H(\boldsymbol{x}).
\end{aligned} \tag{1.14}$$

We call this H a multivariate generalised Pareto distribution (Beirlant, 2004; Rootzén and Tajvidi, 2006). To ensure this definition is valid, $G(\mathbf{0})$ must lie in $(0,1)$, i.e. $G_j(0) > 0$ and $G_j(0) < 1$ for at least one $j \in 1,\dots,d$. For convenience in defining the parameters of H, we may strengthen this condition slightly by requiring $G_j(0) > 0$ for all $j \in 1,\dots,d$. Such restrictions can always be met through an appropriate choice of $\boldsymbol{\alpha}_n$ and $\boldsymbol{\beta}_n$.

In the above definition, H is closely linked to an mGEV distribution G. Let $\boldsymbol{\mu},\boldsymbol{\alpha},\boldsymbol{\gamma}$ denote the location, scale, and shape parameters of the marginal GEVs, and ℓ to represent the stdf. If we parameterise G by $(\boldsymbol{\mu},\boldsymbol{\alpha},\boldsymbol{\gamma},\ell)$, then H inherits this parameterisation. However, just like the univariate GPD in Theorem 3, where we lose one free parameter when deriving a GPD from a GEV distribution, $\boldsymbol{\mu}$ and $\boldsymbol{\alpha}$ cannot both be identified in H. The reason is that for any $t > 0$, the mGEV distributions G and G^t produce the same H. Owing to the max-stability of G, G^t can be obtained by applying a location–scale transformation to G; hence, they differ only in the marginal location and scale parameters. If we parameterise G^t as $(\boldsymbol{\mu}(t),\boldsymbol{\alpha}(t),\boldsymbol{\gamma},\ell)$, then it is easy to show

$$\begin{aligned}\boldsymbol{\mu}(t) &= \boldsymbol{\mu} + \boldsymbol{\alpha}\left(t^{\gamma} - 1\right)/\gamma, \\ \boldsymbol{\alpha}(t) &= t^{\gamma}\boldsymbol{\alpha}.\end{aligned}$$

by applying power transformation on marginal GEVs in (1.2) and comparing the parameters. Notice that the equality

$$\boldsymbol{\alpha} - \boldsymbol{\gamma}\boldsymbol{\mu} = \boldsymbol{\alpha}(t) - \boldsymbol{\gamma}\boldsymbol{\mu}(t)$$

holds for all $t > 0$. This indicates H can be parameterised in terms of $\boldsymbol{\sigma} = \boldsymbol{\alpha} - \boldsymbol{\gamma}\boldsymbol{\mu}$ to remove the identification issue, which aligns with the form of scale parameter in GPD in Theorem 3. Furthermore, $\boldsymbol{\sigma}$ can be interpreted as the scale parameter, since $\sigma_j = \alpha_j - \gamma_j > 0$ is guaranteed by $G_j(0) > 0$. With this reparameterisation, an mGPD has identifiable marginal parameters $\boldsymbol{\sigma}$ (scale) and $\boldsymbol{\gamma}$ (shape).

We can further explicitly express H in terms of $\boldsymbol{\sigma}$, $\boldsymbol{\gamma}$ and ℓ by substituting (1.12) into (1.14):

$$\begin{aligned}H(\boldsymbol{x}) =& \frac{1}{\log G(\mathbf{0})} \log \frac{G(\boldsymbol{x} \wedge \mathbf{0})}{G(\boldsymbol{x})} \\ =& \frac{1}{\log G(\mathbf{0})} \{\ell(\log G_1(x_1 \wedge 0), \ldots, \log G_d(x_d \wedge 0)) - \ell(\log G_1(x_1), \ldots, \log G_d(x_d))\} \\ =& \ell\left\{\boldsymbol{w}\left(1 + \gamma\frac{\boldsymbol{x} \wedge \mathbf{0}}{\boldsymbol{\sigma}}\right)^{-1/\gamma}\right\} - \ell\left\{\boldsymbol{w}\left(1 + \gamma\frac{\boldsymbol{x}}{\boldsymbol{\sigma}}\right)^{-1/\gamma}\right\},\end{aligned} \tag{1.15}$$

where $\boldsymbol{w} = (w_1, \ldots, w_d)$, $w_j = \{\log G_j(0)\}/\{\log G(\mathbf{0})\}$. The last line follows from dividing each argument of ℓ by $\log G_j(0)$ and compensating with the corresponding factor, while the homogeneity of ℓ permits $\log G(\mathbf{0})$ to be taken inside. The vector $\boldsymbol{w}$ consists of the ratios $\log G_j(0)/\log G(\mathbf{0})$ and therefore carries certain dependence structure information. By definition, $\boldsymbol{w} \in (\mathbf{0}, \mathbf{1}]$ and satisfies the normalisation condition $\ell(\boldsymbol{w}) = 1$. As Rootzén et al. (2018b) pointed out, H can be fully determined if $\boldsymbol{\sigma}$, $\boldsymbol{\gamma}$, $\boldsymbol{w}$ and ℓ are all known . We therefore denote this distribution by $\mathrm{mGPD}(\boldsymbol{\sigma}, \boldsymbol{\gamma}, \boldsymbol{w}, \ell)$.

Before examining the dependence structure of the mGPD, we highlight some elegant properties that follow directly from the definition (1.14). These results can be established using only the mGPD definition and the max-stability of G.

Proposition 6 (Kiriliouk et al. (2019))**.** *Suppose $\boldsymbol{X}$ is a d-dimensional random vector and $\boldsymbol{X} \sim \mathrm{mGPD}(\boldsymbol{\sigma}, \boldsymbol{\gamma}, \boldsymbol{w}, \ell)$ with its CDF denoted as H, then the following properties hold.*

1. ***Threshold Stability**. For a threshold $\boldsymbol{u} \geq \boldsymbol{0}$ satisfying $H(\boldsymbol{u}) < 1$ and $\boldsymbol{\sigma} + \boldsymbol{\gamma}\boldsymbol{u} > \boldsymbol{0}$,*

$$\boldsymbol{X} - \boldsymbol{u} \mid \boldsymbol{X} \nleq \boldsymbol{u} \sim \mathrm{mGPD}(\boldsymbol{\sigma} + \boldsymbol{\gamma}\boldsymbol{u}, \boldsymbol{\gamma}, \boldsymbol{w}, \ell).$$

2. ***Lower dimensional conditional margins**. For any $J \subset 1, \ldots, d$, $\boldsymbol{X}_J \mid \boldsymbol{X}_J \nleq \boldsymbol{0}_J$ is a mGPD.*

The threshold stability result implies that increasing the threshold of an mGPD leads to another mGPD, differing only in the marginal scale parameters. As $\boldsymbol{w}$ and ℓ remain unchanged with threshold choice, tail dependence metrics based on them are constant for an mGPD. This property is useful for threshold selection in threshold exceedance modelling, a point to be discussed later.

For the second property, in the special case $J = j$, the conditional distribution $X_j \mid X_j > 0$ is univariate GPD, thereby revealing the connection between univariate and multivariate definitions. One caveat is that the unconditional distribution of X_J is not, in general, an mGPD (Kiriliouk et al., 2019).

Now, the only thing we need to do is to specify $\boldsymbol{w}$ and ℓ. Equation (1.15) suggests a transformation to simplify the form of H

$$\boldsymbol{Z} = \mathbb{1}\{\boldsymbol{\gamma} \neq \boldsymbol{0}\}\frac{1}{\boldsymbol{\gamma}}\log\left(1 + \boldsymbol{\gamma}\frac{\boldsymbol{X}}{\boldsymbol{\sigma}}\right) + \mathbb{1}\{\boldsymbol{\gamma} = \boldsymbol{0}\}\frac{\boldsymbol{X}}{\boldsymbol{\sigma}}. \tag{1.16}$$

Intuitively, this transformation standardises each conditional margin $X_j \mid X_j > 0$ to a standard univariate GPD with scale parameter $\sigma_j = 1$ and shape parameter $\gamma_j = 0$. We denote $\mathrm{mGPD}(\boldsymbol{1}, \boldsymbol{0}, \boldsymbol{w}, \ell)$ as the standard mGPD, with CDF $H(\boldsymbol{z})$ and density $h(\boldsymbol{z})$.

One approach to developing a parametric form for $\mathrm{mGPD}(\boldsymbol{1}, \boldsymbol{0}, \boldsymbol{w}, \ell)$ is to use directly the $\boldsymbol{w}$ and ℓ derived from an exponent function (e.g. (1.10)). However, this method only yields the CDF and density of an mGPD, while simulation from the distribution is not straightforward. We can actually have both a simple sampling algorithm and an explicit density function of an mGPD using its stochastic representation.

Recall that a function $\ell : [\boldsymbol{0}, \boldsymbol{\infty}) \to [0, \infty]$ is a valid stable tail dependence function (stdf) if there exists a random vector $\boldsymbol{V} \geq \boldsymbol{0}$ with $\mathbb{E}(V_j) = 1$ for $j = 1, \ldots, d$. Such a vector $\boldsymbol{V}$ can be constructed from another random vector $\boldsymbol{S}$ by

$$\boldsymbol{V} = \frac{\exp\{\boldsymbol{S}\}}{\mathbb{E}(\exp\{\boldsymbol{S}\})}$$

If we further assume $\boldsymbol{S}$ takes values in $[-\infty, \mathbf{0}]$ and satisfies

$$\begin{aligned} &1. \quad \mathbb{P}(\max(S_1, \ldots, S_d) = 0) = 1 \\ &2. \quad \mathbb{P}(S_j > -\infty) > 0 \qquad \text{for all } j = 1, \ldots, d \end{aligned} \tag{1.17}$$

then $\boldsymbol{S}$ is called a spectral random vector. The following proposition establishes the link between an mGPD and a spectral random vector.

Proposition 7 (Rootzén et al. (2018b)). *For any* $\mathrm{mGPD}(\mathbf{1}, \mathbf{0}, \boldsymbol{w}, \ell)$ *with* $\boldsymbol{w} \in (\mathbf{0}, \mathbf{1}]$ *and stdf* $\ell(\boldsymbol{w}) = 1$*, there exist a spectral random vector* $\boldsymbol{S}$ *unique in distribution, and a unit exponent exponent random vector* E *independent of* $\boldsymbol{S}$*, such that*

$$\boldsymbol{S} + E \sim \mathrm{mGPD}(\mathbf{1}, \mathbf{0}, \boldsymbol{w}, \boldsymbol{\ell}).$$

$\boldsymbol{w}$*,* ℓ*, CDF and density functions are given by*

$$\begin{aligned} w_j &= \mathbb{E}(\exp\{S_j\}), \quad j = 1, \ldots, d; \\ \ell(\boldsymbol{z}) &= \mathbb{E}\left(\max\left(\frac{\boldsymbol{z}\exp\{\boldsymbol{S}\}}{\boldsymbol{w}}\right)\right); \\ H(\boldsymbol{z}) &= 1 - \mathbb{E}[\min(1, e^{\max(\boldsymbol{S}-\boldsymbol{z})})]; \\ h(\boldsymbol{z}) &= \mathbb{1}\{\boldsymbol{z} \not\leq \mathbf{0}\} f_{\boldsymbol{S}}(\boldsymbol{z} - \max(\boldsymbol{z})) \exp\{-\max(\boldsymbol{z})\}, \end{aligned}$$

where $f_{\boldsymbol{S}}$ *is the Lebesgue density of* $\boldsymbol{S}$*.*

Proposition 7 is crucial: it elevates the mGPD from a theoretical construct to a practical modelling tool. It not only supplies a direct sampling method but also reconstructs the CDF and density in terms of $\boldsymbol{S}$, which is often more intuitive than working with the spectral measure. Building on this, three equivalent representations facilitate practical implementation of the mGPD.

1. **T-Representation**. The idea of the T-representation is to relax the restrictions on the spectral random vectors. Let $\boldsymbol{T}$ be a random vector taking values in $[-\infty, \infty)$ and satisfy the following two mild conditions

   (a) $\mathbb{P}(T_j > -\infty) > 0$ for all $j = 1, \ldots, d$;

   (b) $\mathbb{P}(\max(\boldsymbol{T}) > -\infty) = 1$.

Then, a spectral random vector $\boldsymbol{S}$ can be represented as

$$\boldsymbol{S} = \boldsymbol{T} - \max(\boldsymbol{T}).$$

The quantities in Proposition 7 are then rewritten in terms of $\boldsymbol{T}$ as:

$$\begin{aligned}
w_j &= \mathbb{E}(\exp\{T_j - \max(\boldsymbol{T})\}), \quad j = 1, \ldots, d;\\
\ell(\boldsymbol{z}) &= \mathbb{E}\left(\max\left(\frac{\boldsymbol{z}\exp\{\boldsymbol{T} - \max(\boldsymbol{T})\}}{\boldsymbol{w}}\right)\right);\\
H(\boldsymbol{z}) &= 1 - \mathbb{E}[\min(1, e^{\max(\boldsymbol{T}-\boldsymbol{z})-\max(\boldsymbol{T})})];\\
h(\boldsymbol{z}) &= \mathbb{1}\{\boldsymbol{z} \not\leq \boldsymbol{0}\}\frac{1}{\exp\{\max(\boldsymbol{z})\}}\int_0^\infty f_{\boldsymbol{T}}(\boldsymbol{z} + \log t)t^{-1}\mathrm{d}t,
\end{aligned}$$

2. **R-Representation** The R-representation arises from a point-process perspective of the mGEV distribution. Let $\boldsymbol{\sigma} \in (0,\infty)^d$, $\boldsymbol{\gamma} \in \mathbb{R}^d$, and let $\boldsymbol{R}$ be a random vector with support D and distribution function $F_{\boldsymbol{R}}$, where

$$D = I_1 \times \cdots \times I_d, \quad I_j = \begin{cases} [0,\infty) & \text{if } \gamma_j > 0, \\ (-\infty,\infty) & \text{if } \gamma_j = 0, \\ (-\infty, 0] & \text{if } \gamma_j < 0, \end{cases} \quad j = 1, \ldots, d. \tag{1.18}$$

Assume $0 < \mathbb{E}(|R_j|^{1/\gamma_j}) < \infty$ if $\gamma_j \neq 0$ and $\mathbb{E}(\exp\{R_j/\sigma_j\}) < \infty$ if $\gamma_j = 0$. Let $(T_i)_{i\geq 1}$ be the points of a unit-rate Poisson process on $(0,\infty)$, independent of an i.i.d. sequence $(\boldsymbol{R}_i)_{i\geq 1}$ with common law $F_{\boldsymbol{R}}$. Thus the point process $w = \sum_{i\geq 1}\delta_{(T_i,\boldsymbol{R}_i)}$ on $(0,\infty)\times D$ has intensity $\mu(\mathrm{d}t, \mathrm{d}\boldsymbol{r}) = \mathrm{d}tF_{\boldsymbol{R}}(\mathrm{d}\boldsymbol{r})$. Define mapping $\phi : (0,\infty)\times D \to \mathbb{R}^d$:

$$\phi(t,\boldsymbol{r}) = \frac{\boldsymbol{r}}{t^{\boldsymbol{\gamma}}} - \frac{\boldsymbol{\sigma}}{\boldsymbol{\gamma}} \qquad \left(\text{with } \frac{r_j}{t^{\gamma_j}} - \frac{\sigma_j}{\gamma_j} \text{ interpreted as } r_j - \sigma_j \log t \text{ if } \gamma_j = 0\right),$$

and a point process $N = \sum_{i\geq 1}\delta_{\phi(T_i,\boldsymbol{R}_i)}$ on $\mathbb{R}^d$. Its intensity measure Λ is the pushforward $\mu \circ \phi^{-1}$, i.e.,

$$\Lambda(B) = \int_0^\infty \int_D \mathbb{1}\left\{\phi(t,\boldsymbol{r}) \in B\right\} F_{\boldsymbol{R}}(\mathrm{d}\boldsymbol{r})\,\mathrm{d}t, \qquad B \subset \mathbb{R}^d \text{ Borel},$$

Let $A_{\boldsymbol{x}} = \{\boldsymbol{y} \in \mathbb{R}^d : \boldsymbol{y} \not\leq \boldsymbol{x}\}$, then the distribution

$$\begin{aligned} G(\boldsymbol{x}) :=& \mathbb{P}\left(\sup_{i\geq 1} \phi(T_i, \boldsymbol{R}_i) \leq \boldsymbol{x}\right) \\ =& \mathbb{P}\left(N(A_{\boldsymbol{x}}) = 0\right) \\ =& \exp\{-\Lambda(A_{\boldsymbol{x}})\} \\ =& \exp\left\{-\int_0^\infty \bar{F}_{\boldsymbol{R}}\left(t^{\boldsymbol{\gamma}}\left(\boldsymbol{x} + \frac{\boldsymbol{\sigma}}{\boldsymbol{\gamma}}\right)\right) \mathrm{d}t\right\}, \end{aligned} \tag{1.19}$$

is an mGEV distribution (Rootzén et al., 2018a). Here, $\bar{F}_{\boldsymbol{R}} = 1 - F_{\boldsymbol{R}}$, and $t^{\gamma_j}(x_j + \sigma_j/\gamma_j)$ is interpreted as $x_j + \sigma_j \log t$ when $\gamma_j = 0$. Conversely, for any mGEV distribution G, there exists an $\boldsymbol{R}$ such that $G(\boldsymbol{x}) = \mathbb{P}(\sup_{i\geq 1} \phi(T_i, \boldsymbol{R}_i) \leq \boldsymbol{x})$. Substituting (1.19) into (1.14) yields

$$\begin{aligned} H_R(\boldsymbol{x}) &= \frac{\int_0^\infty \left\{F_{\boldsymbol{R}}\left(t^{\boldsymbol{\gamma}}(\boldsymbol{x} + \boldsymbol{\sigma}/\boldsymbol{\gamma})\right) - F_{\boldsymbol{R}}\left(t^{\boldsymbol{\gamma}}((\boldsymbol{x} \wedge \boldsymbol{0}) + \boldsymbol{\sigma}/\boldsymbol{\gamma})\right)\right\} \mathrm{d}t}{\int_0^\infty \bar{F}_{\boldsymbol{R}}\left(t^{\boldsymbol{\gamma}}\boldsymbol{\sigma}/\boldsymbol{\gamma}\right) \mathrm{d}t}, \\ h(\boldsymbol{x}) &= \mathbb{1}(\boldsymbol{x} \not\leq \boldsymbol{0}) \frac{1}{\mathbb{E}\left(\max\{(\boldsymbol{\gamma}\boldsymbol{R}/\boldsymbol{\sigma})^{1/\boldsymbol{\gamma}}\}\right)} \int_0^\infty f_{\boldsymbol{R}}\left(t^{\boldsymbol{\gamma}}(\boldsymbol{x} + \boldsymbol{\sigma}/\boldsymbol{\gamma})\right) t^{\sum_{j=1}^d \gamma_j} \mathrm{d}t \end{aligned}$$

3. **U-Representation** For a random vector $\boldsymbol{R}$ as in the R-representation, define

$$\boldsymbol{U} = \frac{1}{\boldsymbol{\gamma}} \log\left(\frac{\boldsymbol{\gamma}\boldsymbol{R}}{\boldsymbol{\sigma}}\right),$$

with components taking the limit form when $\gamma_j = 0$. This change of variable leads to the U-representation:

$$H_U(\boldsymbol{x}) = \frac{\int_0^\infty \left\{F_{\boldsymbol{U}}\left(\frac{1}{\gamma}\log\left(\frac{\gamma}{\sigma}\boldsymbol{x} + 1\right) + \log t\right) - F_{\boldsymbol{U}}\left(\frac{1}{\gamma}\log\left(\frac{\gamma}{\sigma}(\boldsymbol{x} \wedge \boldsymbol{0}) + 1\right) + \log t\right)\right\} \mathrm{d}t}{\int_0^\infty \bar{F}_{\boldsymbol{U}}(\log t)\mathrm{d}t}$$

where $F_{\boldsymbol{U}}$ is the CDF of $\boldsymbol{U}$ and $\bar{F}_{\boldsymbol{U}} = 1 - F_{\boldsymbol{U}}$. Using the transformation in (1.16), this simplifies to

$$\begin{aligned} H(\boldsymbol{z}) &= \frac{\int_0^\infty \left\{F_{\boldsymbol{U}}(\boldsymbol{z} + \log t) - F_{\boldsymbol{U}}(\boldsymbol{z} \wedge \boldsymbol{0} + \log t)\right\} \mathrm{d}t}{\int_0^\infty \bar{F}_{\boldsymbol{U}}(\log t)\mathrm{d}t} \\ &= \frac{\mathbb{E}(\exp\{\max(\boldsymbol{U} \boldsymbol{-} (\boldsymbol{z} \wedge \boldsymbol{0})\} - \exp\{\max(\boldsymbol{U} - \boldsymbol{z})\})}{\mathbb{E}(\exp\{\max(\boldsymbol{U})\})} \\ &= 1 - \frac{\mathbb{E}\left(\exp\{\max(\boldsymbol{U})\} \wedge \exp\{\max(\boldsymbol{U} - \boldsymbol{z})\}\right)}{\mathbb{E}\left(\exp\{\max(\boldsymbol{U})\}\right)}. \end{aligned}$$

Here we use the fact that

$$\begin{aligned}\int_0^\infty \bar{F}_{\boldsymbol{U}}(\log t)\mathrm{d}t &= \int_0^\infty \mathbb{P}(\boldsymbol{U} \nleq \log t)\mathrm{d}t \\ &= \int_0^\infty \mathbb{P}(\exp\{\max(\boldsymbol{U})\} > t)\mathrm{d}t \\ &= \mathbb{E}\left(\exp\{\max(\boldsymbol{U})\}\right)\end{aligned}$$

for the denominator (and a similar derivation for numerators), and identities $\max\{\boldsymbol{U} - (\boldsymbol{z} \wedge \boldsymbol{0})\} = \max(\boldsymbol{U} - \boldsymbol{z}) \vee \max(\boldsymbol{U})$, and $\boldsymbol{a} \vee \boldsymbol{b} - \boldsymbol{a} = \boldsymbol{b} - \boldsymbol{b} \wedge \boldsymbol{a}$. Note that the $H_{\boldsymbol{U}}(\boldsymbol{z})$ has a very similar form compared to $H_{\boldsymbol{S}}(\boldsymbol{z})$. In fact, by comparing these two CDFs, one can define a spectral vector $\boldsymbol{S}$ via $\boldsymbol{U}$ as

$$\mathbb{E}[f(\boldsymbol{S}) \in \cdot] = \frac{\mathbb{E}[\exp\{\max(\boldsymbol{U})\}f(\boldsymbol{U} - \max\{\boldsymbol{U}\}) \in \cdot]}{\mathbb{E}(\exp\{\max(\boldsymbol{U})\})}$$

for any measurable function f. Taking $f(\cdot) = \mathbb{1}\{\cdot\}$ verifies that this $\boldsymbol{S}$ satisfies (1.17). The corresponding density of the U-representation is

$$h(\boldsymbol{z}) = \mathbb{1}(\boldsymbol{z} \nleq \boldsymbol{0})\frac{1}{\mathbb{E}\left(\exp\{\max(\boldsymbol{U})\}\right)}\int_0^\infty f_{\boldsymbol{U}}(\boldsymbol{z} + \log t)\mathrm{d}t.$$

Since the T, U, and R representations are all linked to the spectral random vector $\boldsymbol{S}$, and the law of $\boldsymbol{S}$ is invariant across them, they are equivalent at the distributional level. Practically, it is easier to specify generators $\boldsymbol{T}$, $\boldsymbol{U}$, or $\boldsymbol{R}$ than the bounded $\boldsymbol{S}$. However, the integral in their density expressions restricts generator choices if a closed-form $h(\boldsymbol{z})$ is required. In that sense, the various representations are one way to increase the expressivity of the $h(\boldsymbol{z})$ when rich forms of the generator are unavailable. Kiriliouk et al. (2019) studied several parametric generator families, including multivariate Gaussian, independent components with Gumbel, reverse Gumbel, reverse exponential, log-Gamma, and structured components, all of which yield closed-form $h(\boldsymbol{z})$. We conclude this section by presenting the $h(\boldsymbol{z})$ with independent reverse exponential components in the T-representation. In this case, $f_{\boldsymbol{T}}$ is assumed to be

$$f_{\boldsymbol{T}}(\boldsymbol{u}) = \prod_{j=1}^d \alpha_j \exp\{\alpha_j(u_j + \beta_j)\}, \quad u_j \in (-\infty, -\beta_j),\ \alpha_j > 0,\ \beta_j \in \mathbb{R}.$$

The corresponding density is

$$h(\boldsymbol{z}) = \mathbb{1}(\boldsymbol{z} \nleq \boldsymbol{0}) \frac{\exp\{-\max(\boldsymbol{z}) - \max(\boldsymbol{z}+\boldsymbol{\beta})\sum_{j=1}^{d}\alpha_j\}}{\sum_{j=1}^{d}\alpha_j} \prod_{j=1}^{d} \alpha_j \exp\{\alpha_j(x_j+\beta_j)\}.$$

Tail Dependence

A central question in multivariate extremes is whether extreme events occur simultaneously; in other words, the tail dependence. This arises frequently in environmental applications, both when studying a single process across multiple locations and when analysing multiple processes at a single site (e.g. modelling still-water level and waves for flood detection (Bortot et al., 2000)). For a two-dimensional random vector $\boldsymbol{X} = (X_1, X_2)$ with marginal distributions F_1 and F_2, a widely used measure of tail dependence is the tail dependence coefficient χ (Coles et al., 1999) defined as

$$\chi = \lim_{u \to 1} \chi(u), \qquad \chi(u) = \mathbb{P}(F(X_2) > u \mid F(X_1) > u), \quad u \in (0,1)$$

This definition can be extended to d dimensions by replacing $\chi(u)$ with

$$\begin{aligned}\chi(u) =& \mathbb{P}(F_j(X_j) > u \text{ for all } j \mid F_j(X_j) > u)\\ =& \frac{\mathbb{P}\left(\bigcap_{j=1}^{d}\{F_j(X_j) > u\}\right)}{1-u}, \qquad\qquad u \in (0,1).\end{aligned}$$

We call $\boldsymbol{X}$ asymptotically dependent (AD) if $\chi > 0$, and call it asymptotically independent (AI) when $\chi = 0$. This AD/AI classification is distinct from classical dependence/independence: dependence does not imply asymptotic dependence. For example, Sibuya et al. (1960) showed that for a bivariate normal distribution, one has $\chi = 0$ and hence the distribution is AI when its correlation ρ satisfies $|\rho| < 1$. Tail dependence, therefore, requires separate investigation from bulk dependence.

For an mGPD in Section 1.2, taking the T-representation as an example, and letting $u^* = \max(H_1(0), \ldots, H_d(0))$, one can show that its tail dependence coefficient is

$$\begin{aligned}\chi &= \chi(u; u > u^*)\\ &= \mathbb{E}(\min(\boldsymbol{V}))\\ &= \mathbb{E}\left(\min\left\{\frac{\exp\{T_1 - \max(\boldsymbol{T})\}}{\mathbb{E}(\exp\{T_1 - \max(\boldsymbol{T})\})}, \cdots, \frac{\exp\{T_d - \max(\boldsymbol{T})\}}{\mathbb{E}(\exp\{T_d - \max(\boldsymbol{T})\})}\right\}\right)\end{aligned} \tag{1.20}$$

by applying the inclusion-exclusion formula on the stdf (Rootzén et al., 2018b; Hu and Castro-Camilo, 2025). Two implications follow from (1.20). First, an mGPD is AD regardless of the generator, as the χ is always positive. Consequently, the mGPD is only suitable for data exhibiting AD behaviour, which can be checked via the diagnostic plot of the empirical $\chi(u)$. Second, when $u > u^*$, that is, in the region where all components exceed the threshold, $\chi(u)$ is constant and equals χ. This constancy follows from the threshold-stability of the mGPD and can be exploited to select a suitable threshold in applications. Specifically, we can standardise the margins to the uniform scale using empirical CDFs and plot the empirical $\chi(u)$ over a grid $0 < u_1 < \cdots < u_n < 1$. A suitable threshold could be the $\tilde{u}$-quantile of the margins, where $\tilde{u}$ is the smallest u_i such that the empirical $\chi(u)$ is approximately constant for all $u \geq \tilde{u}$.

1.3 Jointly modelling the bulk and tail

In many applications, both median-level and extreme events are of interest. For example, in modelling wildfires in Portugal, where annual economic losses are estimated at €60–140 million (Government of Portugal, 2021), it is important to capture both mega-fires with large burned areas and the frequent, smaller fires in order to support accurate fire risk management. This motivates modelling the full distribution of the data, with particular attention to the tail.

Since the tail of a distribution can be well approximated by a GPD, a natural idea is to use one distribution for the bulk and splice in a GPD for the tail. If we consider only the right tail (typical in environmental applications due to non-negative observations), the piecewise framework can be sketched as

$$F(x) = \begin{cases} F_{\text{bulk}}(x), & x < u, \\ F_{\text{bulk}}(u) + [1 - F_{\text{bulk}}(u)]F_{\text{GPD}}(x - u), & x \geq u. \end{cases} \tag{1.21}$$

where F_{bulk} is a CDF on $(0, \infty)$ for the bulk component and may be specified parametrically (Behrens et al., 2004), semi-parametrically (do Nascimento et al., 2012), or non-parametrically (MacDonald et al., 2011). The threshold u defines the boundary between the bulk model and the tail model: values below u are described by F_{bulk}, while exceedances above u are modelled using the GPD approximation. An advantage of this framework is that u is treated as a learnable parameter, bypassing manual threshold selection in peak-over-threshold methods. Inference is typically performed in a Bayesian framework, which further provides insight into the uncertainty of u.

A natural extension is to ask whether this bulk–tail joint modelling can also be done in the multivariate setting. Returning to the wildfire example, one may be interested not only in the distribution of wildfires at a single site but also across municipalities, in order to understand spatial patterns. The answer is yes, but three main challenges must be addressed before using multivariate EVT for the tail:

1. **Parametric forms of the tail**. Unlike univariate EVT, where the asymptotic distributions of maxima and threshold exceedances are uniquely defined, multivariate EVT admits infinitely many parameterisations of the dependence structure. The few available parametric families are typically designed to admit closed-form expressions for tractability. This restricts flexibility and can introduce bias if the chosen dependence structure deviates substantially from reality.

2. **Discontinuity in margins and dependence.** The support of the mGPD is a subset of $\mathbb{R}^d$, so combining it with another distribution for the bulk inevitably introduces discontinuities in both margins and dependence. Such discontinuities may be unrealistic when the dataset is large.

3. **Restricted tail dependence**. Classic multivariate EVT is built on the max-stable mGEV distribution. A direct consequence is that the tail dependence is always asymptotically dependent. If we insist on mixing a multivariate EVT distribution with another distribution for the bulk, the resulting framework cannot model cases with asymptotic independence. This is restrictive in high-dimensional applications such as spatial modelling, where pairwise tail dependence may vary with distance between sites, and may exhibit both asymptotic dependence and asymptotic independence.

In this thesis, we propose three methods, ranging from statistical modelling to deep learning, to address these challenges and provide practical frameworks for modelling of bulk and tail simultaneously.

As a first step, we construct a mixed distribution by piecing together a bivariate normal distribution with a bivariate GPD. This can be viewed as a bivariate extension of the work of Behrens et al. (2004) and do Nascimento et al. (2012). The main motivation is to preserve all EVT-guaranteed tail properties while providing a simple framework, which is particularly valuable for small samples. The framework proves effective on small datasets, with performance comparable to alternative threshold-free joint models focused on the dependence modelling (André et al., 2024). However, it does not resolve the three challenges above and thus is limited in high-dimensional, large-sample applications.

To address the infinite-parameterisation issue, we focus on the mGPD and propose representing its dependence structure with normalising flows, a class of deep generative models. This framework, termed GPDFlow, allows the dependence structure of the mGPD to be learned from the data rather than pre-specified. Owing to the high expressivity of normalising flows, GPDFlow can, in principle, approximate any well-behaved dependence structure.

The discontinuity issue can be mitigated if both margins and dependence are continuous. Continuous margins can be obtained using sub-asymptotic distributions, which reproduce EVT-consistent tails, i.e., Fréchet ($\gamma > 0$), Gumbel ($\gamma = 0$), or Weibull ($\gamma < 0$), while tailoring the bulk to the task. One such distribution is the extended generalised Pareto (eGP) distribution (Naveau et al., 2016), which behaves like a GPD in the right tail but resembles a Gamma distribution on the left tail. For the dependence structure, one approach is to use a Gaussian field to represent the latent dependence structure among different components, and parametrise the likelihood function in terms of linear projections of latent Gaussian fields to capture the dependence of the observations implicitly. This leads to the latent Gaussian modelling, which is widely applied in areal modelling where spatial dependence is naturally expressed via adjacency structures such as the conditional autoregressive model. By using the eGP as the likelihood, we obtain a joint bulk-tail model with EVT-compliant margins.

The above latent Gaussian model framework is already able to relax the strong AD assumption of the multivariate EVT while retaining EVT-consistent margins. However, it relies on the fact that latent Gaussian fields can adequately represent the dependence, and this is not always realistic. For example, when modelling precipitation and sea level at a single coastal location, it is hard to find a proper latent Gaussian field to describe the dependence between these two quantities.

To allow more flexibility, we explore whether generative models, particularly the Denoising Diffusion Probabilistic Model, can represent the entire dependence structure and thereby model both bulk and tail jointly. The key distinction from GPDFlow is that here the generative model is applied to the full dataset, not just threshold exceedances. While GPDFlow is necessarily asymptotically dependent, the diffusion model can, in principle, accommodate more flexible tail dependence structures. A common challenge, however, is that generative models often struggle to represent heavy-tailed data (Jaini et al., 2020; Pandey et al., 2024). This can be alleviated either by adapting the model architecture to account for heavy tails, as in GPDFlow, or by transforming the data (e.g. logarithmic or Box–Cox transformations) to reduce tail heaviness prior to model fitting. We adopt the latter approach and examine its effectiveness as our final contribution.

1.4 Outline of the thesis

The remaining part of the thesis is organised as follows.

- Chapter 2 provides the methodological background required to understand the work in this thesis. We first review posterior simulation methods in Bayesian inference, with emphasis on Markov Chain Monte Carlo methods and integrated nested Laplace approximation. We then summarise the core machine learning components that appear in later chapters: gradient boosted decision trees via XGBoost, and deep learning architectures, including multilayer perceptron (MLP), recurrent networks (LSTM) and generative models (normalsing flows, diffusion models).

- Chapter 3 presents the methodology for piecing together a multivariate bulk distribution and an mGPD for the tail. We illustrate the framework in two dimensions, using a bivariate Gaussian distribution for the bulk and a U-representation mGPD with independent reverse exponential components as the generator. This framework shows a competitive performance in dependence modelling compared to a method that focuses on smoothly mixing a copula tailored to the bulk and a copula tailored to the tail (André et al., 2024).

- Chapter 4 proposes GPDFlow, which tackles the infinite-parameterisation issue by representing the mGPD generator via normalising flows. We investigate its tail properties and propose a threshold selection method tailored for partial exceedance probabilities (i.e. when only a subset of components exceeds the threshold). Simulations and an application to systemic risk among five major US banks demonstrate the accuracy and flexibility gains of GPDFlow over classical parametric mGPDs.

- Chapter 5 develops a spatio-temporal joint bulk–tail model using a latent Gaussian framework with an eGP likelihood, applied to wildfire forecasting in Portugal. Forecasting challenges include modelling both moderate and extreme burned areas, handling complex covariate effects, and managing the absence of future environmental covariates. We propose a two-stage ensemble: a gradient boosting model first identifies wildfire patterns and generates pseudo-covariates, which are then used in a latent Gaussian model with eGP likelihood. Inference is carried out via integrated nested Laplace approximation, with a detailed discussion of penalised-complexity priors for eGP parameters. Our framework effectively addresses the existing challenges in the spatio-temporal forecast and accurately describes the monthly total fire count and burnt area.

- Chapter 6, motivated by the Extreme Value Data Challenge 2025, explores a fully deep learning approach to extrapolate event probabilities over a 5×5 grid of daily precipitation data spanning 165 years. We combine an LSTM to capture temporal dependence with a denoising diffusion probabilistic model for spatial dependence among grid cells, conditional on the LSTM outputs. Since precipitation is non-negative, a logarithmic transformation with zero-adjustment is applied to reduce tail heaviness, enabling the diffusion model to better represent the extremes. Diagnostics on dependence and tail behaviour demonstrate the accuracy of this framework.

- Chapter 7 concludes the thesis and outlines potential directions for integrating deep learning and EVT.

Chapter 2

Methodological framework

In this chapter, we introduce the statistical inference techniques, as well as several machine learning and deep learning models, that will be used in later chapters.

2.1 Methods for Bayesian inference

In a Bayesian statistical model, the unknown parameter vector $\boldsymbol{\theta}$ is treated not as a fixed value, but as a random vector with a probability distribution $\pi(\boldsymbol{\theta})$. This distribution, known as the prior, represents our a priori belief about $\boldsymbol{\theta}$ before observing any data. Once the data vector $\boldsymbol{y}$ is observed, the prior can be updated via Bayes' rule:

$$\pi(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{\pi(\boldsymbol{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\pi(\boldsymbol{y})} = \frac{\pi(\boldsymbol{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int \pi(\boldsymbol{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}},$$

where $\pi(\boldsymbol{y} \mid \boldsymbol{\theta})$ is the likelihood specified by the statistical model. The distribution $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$ is called the posterior and is the key quantity for inference on $\boldsymbol{\theta}$. In most problems, the posterior does not have a closed-form as the normalising constant in the denominator is analytically intractable, so approximation methods are required.

2.1.1 Metropolis–Hastings algorithm

The Metropolis–Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) is one of the most widely used methods for approximating a posterior distribution. It is a Markov chain Monte Carlo (MCMC) method: it generates samples by constructing a Markov chain whose stationary (equilibrium) distribution is the target distribution, and then simulating the chain.

A sequence of random vectors $\boldsymbol{\Theta}^{(1)}, \boldsymbol{\Theta}^{(2)}, \ldots, \boldsymbol{\Theta}^{(T)}$ in $\mathbb{R}^d$ is a first-order Markov chain if, for all $t \in \{1, \ldots, T-1\}$, the distribution of $\boldsymbol{\Theta}^{(t+1)}$ conditional on the past depends only on $\boldsymbol{\Theta}^{(t)}$:

$$\mathbb{P}\big(\boldsymbol{\Theta}^{(t+1)} = \boldsymbol{\theta}^{(t+1)} \mid \boldsymbol{\Theta}^{(1)} = \boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\Theta}^{(t)} = \boldsymbol{\theta}^{(t)}\big) = \mathbb{P}\big(\boldsymbol{\Theta}^{(t+1)} = \boldsymbol{\theta}^{(t+1)} \mid \boldsymbol{\Theta}^{(t)} = \boldsymbol{\theta}^{(t)}\big).$$

Such a chain can be specified by an initial distribution and a transition probability $\mathbb{P}\big(\boldsymbol{\Theta}^{(t+1)} = \boldsymbol{\theta}^{(t+1)} \mid \boldsymbol{\Theta}^{(t)} = \boldsymbol{\theta}^{(t)}\big)$ for $t = 1, \ldots, T-1$. If the transition probabilities do not depend on t, the chain is homogeneous. In that case, the transition kernel can be written as

$$T(\boldsymbol{\theta}^*, \boldsymbol{\theta}) = \mathbb{P}\big(\boldsymbol{\Theta}^{(t+1)} = \boldsymbol{\theta} \mid \boldsymbol{\Theta}^{(t)} = \boldsymbol{\theta}^*\big),$$

for all $\boldsymbol{\theta}, \boldsymbol{\theta}^*$ in the state space and all t.

For a discrete state space, the marginal distribution evolves according to

$$\mathbb{P}(\boldsymbol{\Theta}^{(t+1)} = \boldsymbol{\theta}) = \sum_{\boldsymbol{\theta}^*} \mathbb{P}(\boldsymbol{\Theta}^{(t)} = \boldsymbol{\theta}^*) T(\boldsymbol{\theta}^*, \boldsymbol{\theta}).$$

Let $p(\boldsymbol{\theta}) = \mathbb{P}(\boldsymbol{\Theta}^{(t)} = \boldsymbol{\theta})$ denote the marginal distribution at time t. If applying one transition leaves the distribution unchanged, i.e.,

$$\mathbb{P}(\boldsymbol{\Theta}^{(t+1)} = \boldsymbol{\theta}) = \mathbb{P}(\boldsymbol{\Theta}^{(t)} = \boldsymbol{\theta}) = p(\boldsymbol{\theta}),$$

then $p(\cdot)$ is called invariant (or stationary). A sufficient condition for invariance is the detailed balance condition:

$$p(\boldsymbol{\theta}) T(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = p(\boldsymbol{\theta}^*) T(\boldsymbol{\theta}^*, \boldsymbol{\theta}).$$

Indeed, under detailed balance,

$$\mathbb{P}(\boldsymbol{\Theta}^{(t+1)} = \boldsymbol{\theta}) = \sum_{\boldsymbol{\theta}^*} p(\boldsymbol{\theta}^*) T(\boldsymbol{\theta}^*, \boldsymbol{\theta}) = \sum_{\boldsymbol{\theta}^*} p(\boldsymbol{\theta}) T(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = p(\boldsymbol{\theta}) \sum_{\boldsymbol{\theta}^*} p(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}) = p(\boldsymbol{\theta}).$$

Under mild conditions on the invariant distribution and the transition probabilities, the marginal distribution of a homogeneous Markov chain converges to a unique invariant distribution, regardless of the initial state (Bishop and Nasrabadi, 2006). In this case, the chain is called ergodic, and the invariant distribution is also referred to as the stationary (or equilibrium) distribution.

The key idea behind the Metropolis–Hastings algorithm is to construct a transition kernel whose stationary distribution is a desired target distribution, denoted $\pi(\boldsymbol{\theta})$. Starting from the current state $\boldsymbol{\theta}^{(t)}$, we first draw a proposal $\boldsymbol{\theta}' \sim q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(t)})$, where $q(\cdot \mid \cdot)$ is a proposal distribution. In general, the proposal move alone does not satisfy detailed balance, because typically $\pi(\boldsymbol{\theta}^{(t)})q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(t)}) \neq \pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}')$. To correct for this, we introduce an acceptance probability $A(\cdot,\cdot)$, with $0 \leq A(\cdot,\cdot) \leq 1$, such that

$$\pi(\boldsymbol{\theta}^{(t)})q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(t)})A(\boldsymbol{\theta}',\boldsymbol{\theta}^{(t)}) = \pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}')A(\boldsymbol{\theta}^{(t)},\boldsymbol{\theta}').$$

This yields a transition kernel of the form

$$T(\boldsymbol{\theta}^{(t)},\boldsymbol{\theta}') := q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(t)})A(\boldsymbol{\theta}',\boldsymbol{\theta}^{(t)}).$$

With this choice, the resulting transition kernel satisfies detailed balance with respect to π. Operationally, sampling from the resulting kernel is implemented via an accept–reject step, as in a rejection sampling: draw $\boldsymbol{\theta}'$ from $q(\cdot \mid \boldsymbol{\theta}^{(t)})$, then accept it with probability $A(\boldsymbol{\theta}',\boldsymbol{\theta}^{(t)})$; otherwise, keep the current state.

To determine a suitable form for $A(\boldsymbol{\theta}',\boldsymbol{\theta}^{(t)})$, consider

$$\frac{A(\boldsymbol{\theta}',\boldsymbol{\theta}^{(t)})}{A(\boldsymbol{\theta}^{(t)},\boldsymbol{\theta}')} = \frac{\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^{(t)})q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(t)})} := r(\boldsymbol{\theta}^{(t)},\boldsymbol{\theta}'). \tag{2.1}$$

By symmetry, one may take $A(\boldsymbol{\theta}',\boldsymbol{\theta}^{(t)}) = g(r(\boldsymbol{\theta}^{(t)},\boldsymbol{\theta}'))$ for some function $g : [0,\infty) \to [0,1]$. (2.1) implies the functional requirement $g(r) = rg(1/r)$ for all $r > 0$. A standard choice is $g(r) = \min(1,r)$, which yields

$$A(\boldsymbol{\theta}',\boldsymbol{\theta}^{(t)}) = \min\left\{1, \frac{\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^{(t)})q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(t)})}\right\}$$

The resulting Metropolis–Hastings algorithm is summarised in Algorithm 1.

2.1.2 Gibbs sampling

Gibbs sampling (Geman and Geman, 1984; Gelfand and Smith, 1990) is a Markov chain Monte Carlo (MCMC) method that is particularly useful when the full conditional distributions of the target distribution are available in a form that is easy to sample from.

Let $\boldsymbol{\Theta} = (\Theta_1,\ldots,\Theta_d)$ be a d-dimensional random variable with target density $\pi(\theta_1,\ldots,\theta_d)$. For each coordinate $j \in \{1,\ldots,d\}$, write $\boldsymbol{\theta}_{-j}$ for the vector of all components except the jth

Algorithm 1 Metropolis–Hastings Algorithm

```
Input: unnormalised target density $\pi(\cdot)$; proposal distribution $q(\cdot \mid \cdot)$
Initialise: choose $\boldsymbol{\theta}^{(1)}$ such that $\pi(\boldsymbol{\theta}^{(1)}) > 0$
for $t = 1, 2, \dots$ do
    Sample proposal $\boldsymbol{\theta}' \sim q(\cdot \mid \boldsymbol{\theta}^{(t)})$
    Compute
        $$A(\boldsymbol{\theta}', \boldsymbol{\theta}^{(t)}) \;=\; \min\left\{1,\; \frac{\pi(\boldsymbol{\theta}')\, q(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^{(t)})\, q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(t)})}\right\}$$
    Sample $u \sim \mathrm{Uniform}(0, 1)$
    if $u < A(\boldsymbol{\theta}', \boldsymbol{\theta}^{(t)})$ then
        $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}'$
    else
        $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)}$
    end if
end for
```

one. The full conditional density of Θ_j is then

$$\pi(\theta_j \mid \boldsymbol{\theta}_{-j}) := \pi(\theta_j \mid \theta_1, \dots, \theta_{j-1}, \theta_{j+1}, \theta_d).$$

Gibbs sampling generates a Markov chain by repeatedly updating one component at a time: at each step, it samples Θ_j from its full conditional distribution given the current values of all other components. Cycling through $j = 1, \dots, d$ produces one full Gibbs update. This procedure is summarised in Algorithm 2.

The validity of Gibbs sampling can be understood by viewing each coordinate update as a Metropolis–Hastings step in which the proposal distribution is the corresponding full conditional $\pi(\theta_j \mid \boldsymbol{\theta}_{-j})$. Because this proposal exactly matches the target conditional distribution, every proposed update is accepted, so the acceptance probability is identically 1.

Algorithm 2 Gibbs Sampling Algorithm

```
Input: unnormalised target density $\pi(\cdot)$ with full conditionals $\pi(\theta_j \mid \boldsymbol{\theta}_{-j})$, for $j = 1, \dots, d$
Initialise: choose $\boldsymbol{\theta}^{(1)}$ such that $\pi(\boldsymbol{\theta}^{(1)}) > 0$
for $t = 1, 2, \dots$ do
    $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}^{(t)}$
    for $j = 1, \dots, d$ do
        Sample $\theta'_j \sim \pi(\,\cdot \mid \boldsymbol{\theta}'_{-j})$
        Set $\theta'_j \leftarrow \theta'_j$
    end for
    $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}'$
end for
```

2.1.3 Slice sampling

Suppose the goal is to sample $\boldsymbol{\theta} \in \mathbb{R}^d$ from a target density $\pi(\boldsymbol{\theta})$. Instead of sampling directly from π, slice sampling (Neal, 2003) introduces an auxiliary variable h and considers the augmented space $(\boldsymbol{\theta}, h)$. The main idea is to sample uniformly from the region under the (unnormalised) density curve, $\mathcal{R} = \{(\boldsymbol{\theta}, h) : 0 < h < \pi(\boldsymbol{\theta})\}$. If $(\boldsymbol{\theta}, h)$ is uniformly distributed over $\mathcal{R}$, then the marginal distribution of $\boldsymbol{\theta}$ is proportional to $\pi(\boldsymbol{\theta})$. Therefore, we can obtain samples from the target distribution by drawing from $\mathcal{R}$ and discarding h.

Direct uniform sampling from $\mathcal{R}$ is generally difficult, but it becomes straightforward if we alternate between the two full conditional distributions, i.e. performing a Gibbs sampler on the augmented space. Consider the joint density

$$p(\boldsymbol{\theta}, h) \propto \mathbb{1}\{0 < h < \pi(\boldsymbol{\theta})\},$$

Conditioning on $\boldsymbol{\theta}$, the auxiliary variable is uniform on $(0, \pi(\boldsymbol{\theta}))$:

$$h \mid \boldsymbol{\theta} \sim \text{Uniform}(0, \pi(\boldsymbol{\theta})). \tag{2.2}$$

Conditioning on h, $\boldsymbol{\theta}$ is is uniform over the horizontal slice $S(h) = \{\boldsymbol{\theta} : \pi(\boldsymbol{\theta}) > h\}$, so that

$$\boldsymbol{\theta} \mid h \sim \text{Uniform}(S(h)). \tag{2.3}$$

In one dimension $d = 1$, sampling from $\mathrm{Uniform}(S(h))$ can be implemented by finding an interval that contains the current θ and lies largely within the slice. In higher dimensions ($d > 1$), one often uses a rectangle or hyperrectangle that covers the current point and captures a substantial portion of the slice (Neal, 2003). These alternating updates constitute the slice sampling method, outlined in Algorithm 3.

Algorithm 3 Slice Sampling Algorithm

```
Input: unnormalised target density $\pi(\cdot)$
Initialise: choose $\boldsymbol{\theta}^{(1)}$ such that $\pi(\boldsymbol{\theta}^{(1)}) > 0$
for $t = 1, 2, \ldots$ do
    Sample $h \sim \text{Uniform}\big(0, \pi(\boldsymbol{\theta}^{(t)})\big)$
    Define the slice region
        $S(h) = \{\boldsymbol{\theta} : \pi(\boldsymbol{\theta}) > h\}$
    Sample $\boldsymbol{\theta}^{(t+1)} \sim \mathrm{Uniform}\big(S(h)\big)$
end for
```

2.1.4 Automated factor slice sampling

Factor slice sampling is a variant of slice sampling designed to improve efficiency when the components of $\boldsymbol{\theta}$ are strongly (linearly) correlated under the target density $\pi(\boldsymbol{\theta})$. Factor slice sampling updates the state along an orthogonal set of directions that better matches the principal axes of the target, so that each one-dimensional slice update makes meaningful progress along narrow "ridge-like" regions of high probability.

A convenient way to motivate this is through a local Gaussian (quadratic) approximation to the target,

$$\pi^*(\boldsymbol{\theta}) \propto |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu})\right\}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are (approximately) the mean vector and covariance matrix of $\pi(\boldsymbol{\theta})$. Performing the spectral decomposition $\boldsymbol{\Sigma} = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^T$ yields an orthonormal basis $\boldsymbol{E} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d)$, where $\boldsymbol{e}_j$ is the eigenvector associated with the jth eigenvalue. Then $\boldsymbol{\theta}$ can be written in the rotated coordinates as

$$\boldsymbol{\theta} = \boldsymbol{\mu} + \boldsymbol{E}\boldsymbol{\eta} = \boldsymbol{\mu} + \sum_{j=1}^{d} \eta_j \boldsymbol{e}_j,$$

for coefficients $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_d)^T$. Updating $\boldsymbol{\theta}$ in slice sampling can therefore be carried out by updating the rotated coordinates η_j (equivalently, moving along the directions $\boldsymbol{e}_j$), which leverages the dispersion information encoded in $\boldsymbol{E}$. This idea is reflected in Algorithm 4.

Algorithm 4 Factor Slice Sampling Algorithm

```
1:  Input: unnormalised target density $\pi(\cdot)$; dimension $d$; standard basis vectors $\{\boldsymbol{e}_j\}_{j=1}^d$
2:  Initialise: choose $\boldsymbol{\theta}^{(1)}$ such that $\pi(\boldsymbol{\theta}^{(1)}) > 0$
3:  for $t = 1, 2, \ldots$ do
4:      Sample $h \sim \text{Uniform}\big(0, \pi(\boldsymbol{\theta}^{(t)})\big)$
5:      $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}^{(t)}$
6:      for $j = 1, 2, \ldots, d$ do
7:          Sample
                $r_j \sim \text{Uniform}\Big(\big\{r_j : \pi(\boldsymbol{\theta}' + r_j\boldsymbol{e}_j) > h\big\}\Big)$
8:          $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}' + r_j\boldsymbol{e}_j$
9:      end for
10:     $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}'$
11: end for
```

Step 6 in Algorithm 4 can be interpreted as a Gibbs update in the rotated coordinates.

Conditioning on h, we have a uniform distribution over the slice in $\boldsymbol{\eta}$-space,

$$\boldsymbol{\eta} \mid h \sim \mathrm{Uniform}(\{\boldsymbol{\eta} : \pi(\boldsymbol{\mu} + \boldsymbol{E}\boldsymbol{\eta}) > h\}),$$

which is simply a reparameterisation of sampling $\boldsymbol{\theta}$ uniformly from the slice region. The corresponding one-dimensional full conditional for η_j is

$$\eta_j \mid \eta_{-j}, h \sim \text{Uniform}(\{\eta_j : \pi(\boldsymbol{\mu} + \eta_j \boldsymbol{e}_j + \sum_{k \neq j} \eta_k \boldsymbol{e}_k) > h\})$$

and sampling η_j in this way is equivalent to sampling the increment r_j from the set $\{r_j : \pi(r_j \boldsymbol{e}_j + \boldsymbol{\theta}') > h\}$.

Choosing a good orthonormal basis $\boldsymbol{E}$ typically requires some knowledge of the dependence structure of the target. Tibbits et al. (2014) proposes an automated procedure that estimates a suitable basis during an initial tuning phase by repeatedly drawing samples, computing a sample covariance estimate, and updating the implied eigenvector basis until the basis stabilises. One version of this idea is sketched below.

Algorithm 5 Iterative Covariance Update

```
Initialise: $\widehat{\boldsymbol{E}}^{(1)} \leftarrow \boldsymbol{I}$, $\boldsymbol{\Sigma}^{(1)} \leftarrow \boldsymbol{I}$, $t \leftarrow 1$, converged ← False
while not converged do
    Draw $N$ samples using $\widehat{\boldsymbol{E}}^{(t)}$ in Algorithm 4
    Compute $\boldsymbol{\Sigma}^{(t+1)}$ and its eigenvectors $\boldsymbol{E}^{(t+1)}$
    Estimate a rotation matrix $\boldsymbol{R}$ such that $\boldsymbol{\Sigma}^{(t)}\boldsymbol{R} \approx \boldsymbol{\Sigma}^{(t+1)}$
    $t \leftarrow t + 1$
    if $\sum(\boldsymbol{R} - \boldsymbol{I})$ is below a preset tolerance threshold then
        converged ← True
    end if
end while
```

In addition, Tibbits et al. (2014) discusses how to automate the tuning of the slice “width” used when constructing practical approximations to the set $\{\boldsymbol{\theta} : \pi(\boldsymbol{\theta}) > h\}$, for example via Robbins–Monro recursion from the optimisation literature. Together, these automated steps yield the automated factor slice sampler, which aims to reduce the need for prior dependence knowledge and manual tuning.

2.1.5 Integrated nested Laplace approximation

Integrated nested Laplace approximation (INLA) (Rue et al., 2009) is a fast, deterministic method for approximating posterior distributions in a broad class of Bayesian models known as latent Gaussian models. It is designed for settings where the latent variables have a Gaussian prior with a sparse precision structure, enabling efficient numerical computations.

In a Bayesian latent Gaussian model, each observation y_i, $i = 1, \dots, N$, is assumed to be conditionally independent given the linear predictor η_i and hyperparameters $\boldsymbol{\theta}_1$ of the observation model. The response's mean or quantiles are linked to η_i as in the generalised linear model framework, and η_i comprises random and fixed effects that describe the data in an additive way:

$$\eta_i = \beta_0 + \sum_j \beta_j v_{i,j} + \sum_k \omega_{i,k},$$

where $v_{i,j}$ are covariates for the fixed effects, $\omega_{i,k}$ are latent Gaussian random effects, and β_0, β_j are linear coefficients. Using $\boldsymbol{\theta}_2$ to denote all hyperparameters in β_j, $\omega_{i,k}$, the latent field $\boldsymbol{u} = (\beta_0, \beta_1, \dots, \omega_{1,1}, \cdots)$ is assumed to have a Gaussian prior

$$\boldsymbol{u} \mid \boldsymbol{\theta}_2 \sim \mathcal{N}(\mathbf{0}, Q^{-1}(\boldsymbol{\theta}_2)),$$

where $Q(\boldsymbol{\theta}_2)$ is the precision matrix. The linear predictor $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_n)$ can be expressed by $\boldsymbol{u}$ and a sparse design matrix $\boldsymbol{A}$ by

$$\boldsymbol{\eta} = \boldsymbol{A}\boldsymbol{u}.$$

Let $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ and its prior as $\pi(\boldsymbol{\theta})$, the full posterior distribution is

$$\begin{aligned}\pi(\boldsymbol{u}, \boldsymbol{\theta} \mid \boldsymbol{y}) &\propto \pi(\boldsymbol{y} \mid \boldsymbol{A}\boldsymbol{u}, \boldsymbol{\theta})\pi(\boldsymbol{u} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}) \\ &\propto \pi(\boldsymbol{u} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}) \prod_{i=1}^{N} \pi(y_i \mid (\boldsymbol{A}\boldsymbol{u})_i, \boldsymbol{\theta}),\end{aligned}$$

A central idea in INLA is to approximate $\pi(\boldsymbol{u} \mid \boldsymbol{\theta}, \boldsymbol{y})$ by the Laplace approximation $\widetilde{\pi}(\boldsymbol{u} \mid \boldsymbol{\theta}, \boldsymbol{y})$ so that $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$ can be approximated by

$$\widetilde{\pi}(\boldsymbol{\theta} \mid \boldsymbol{y}) \propto \left. \frac{\pi(\boldsymbol{u}, \boldsymbol{\theta}, \boldsymbol{y})}{\widetilde{\pi}(\boldsymbol{u} \mid \boldsymbol{\theta}, \boldsymbol{y})} \right|_{\boldsymbol{u} = \boldsymbol{u}^*(\boldsymbol{\theta})},$$

where $\boldsymbol{u}^*(\boldsymbol{\theta})$ is the mode of $\pi(\boldsymbol{u} \mid \boldsymbol{\theta}, \boldsymbol{y})$, or equivalently, the mode of the joint density in $\boldsymbol{u}$ for fixed $\boldsymbol{\theta}$. Once $\widetilde{\pi}(\boldsymbol{\theta} \mid \boldsymbol{y})$ is available, marginal posteriors of θ_j can be obtained by numerically integrating out the nuisance parameters $\boldsymbol{\theta}_{-j}$:

$$\pi(\theta_j \mid \boldsymbol{y}) = \int \widetilde{\pi}(\boldsymbol{\theta} \mid \boldsymbol{y}) \mathrm{d}\boldsymbol{\theta}_{-j}.$$

Next, taking the Gaussian margins $\widetilde{\pi}\left(u_i \mid \boldsymbol{\theta}, \mathbf{y}\right)$ from $\widetilde{\pi}(\boldsymbol{u}|\boldsymbol{\theta}, \boldsymbol{y})$, the marginal posterior $\pi(u_i \mid \boldsymbol{y})$ is approximated by

$$\widetilde{\pi}\left(u_i \mid \boldsymbol{y}\right) \approx \sum_k \widetilde{\pi}\left(u_i \mid \boldsymbol{\theta}_k, \boldsymbol{y}\right) \widetilde{\pi}\left(\boldsymbol{\theta}_k \mid \boldsymbol{y}\right) \Delta_k,$$

with integration points $\boldsymbol{\theta}_k$ and weights Δ_k. The marginal posterior of the linear predictors $\pi(\eta_i \mid \boldsymbol{y})$ is derived in a similar manner. Starting with the Gaussian approximation $\widetilde{\pi}(\boldsymbol{u} \mid \boldsymbol{\theta}, \boldsymbol{y})$, the conditional posterior $\pi(\eta_i|\boldsymbol{\theta}, \boldsymbol{y})$ is then also approximated as Gaussian. Its mean and variance can be efficiently computed leveraging the linear relationship $\boldsymbol{\eta} = \boldsymbol{A}\boldsymbol{u}$ and some tricks on computing the i-th diagonal element of the inverse of the precision matrix associated with $\widetilde{\pi}(\boldsymbol{u}|\boldsymbol{\theta}, \boldsymbol{y})$ (Van Niekerk et al., 2023). The marginal posterior $\pi(\eta_i|\boldsymbol{y})$ is then approximated by:

$$\widetilde{\pi}\left(\eta_i \mid \boldsymbol{y}\right) \approx \sum_k \widetilde{\pi}\left(\eta_i \mid \boldsymbol{\theta}_k, \boldsymbol{y}\right) \widetilde{\pi}\left(\boldsymbol{\theta}_k \mid \boldsymbol{y}\right) \Delta_k.$$

Finally, Van Niekerk et al. (2023); van Niekerk and Rue (2024) proposed a low-rank correction to the mean of $\widetilde{\pi}(\boldsymbol{u} \mid \boldsymbol{\theta}, \boldsymbol{y})$ using variational Bayes, further improving the approximation of both $\widetilde{\pi}(u_i \mid \boldsymbol{y})$ and $\widetilde{\pi}(\eta_i \mid \boldsymbol{y})$.

2.2 XGBoost

XGBoost (Chen and Guestrin, 2016) is a scalable and efficient gradient boosting framework (Friedman, 2001) designed for structured data. As an ensemble method, it builds a strong predictive model by adding many simple tree models in sequence. Each new tree is fitted to correct the errors made by the current ensemble, so the model improves in an additive, iterative manner.

In a regression setting, a regression tree $f(\boldsymbol{x})$ with input $\boldsymbol{x} \in \mathbb{R}^d$ partitions the feature space into L disjoint regions $R_1, R_2, \cdots, R_L$, known as leaves, through a sequence of binary splits. For any input vector $\boldsymbol{x}_i$, $i = 1, \ldots, N$, the tree returns the leaf weight w_l corresponding to the region R_l that contains $\boldsymbol{x}_i$:

$$f(\boldsymbol{x}_i) = \sum_{l=1}^{L} w_l \mathbb{1}\{\boldsymbol{x}_i \in R_l\},$$

where $\mathbb{1}\{\cdot\}$ is the indicator function. An XGBoost model with M trees predicts

$$\widehat{y}_i = \sum_{m=1}^{M} f_m(\boldsymbol{x}_i), \quad f_m \in \mathcal{F},$$

where $\mathcal{F}$ denotes the space of all possible trees.

XGBoost fits the trees in a forward stagewise fashion. At boosting iteration m, it adds a new tree f_m to minimise the following regularised objective:

$$\mathcal{L}^{(m)} = \sum_{i=1}^{N} \ell(y_i, \widehat{y}_i^{(m-1)} + f_m(\boldsymbol{x}_i)) + \Omega(f_m),$$

where $\ell(\cdot)$ is a differentiable convex loss function, $\widehat{y}_i^{(m-1)} = \sum_{j=1}^{m-1} f_j(\boldsymbol{x}_i)$ is the prediction from the ensemble up to iteration $m-1$, and $\Omega(f_m) = \gamma L + \frac{1}{2}\lambda\|\boldsymbol{w}\|^2$ penalises the model complexity through the number of leaves L and the leaf weights $\boldsymbol{w} = (w_1, \ldots, w_L)$. To improve the optimisation efficiency, the loss is approximated using a second-order Taylor expansion:

$$\mathcal{L}^{(m)} \approx \sum_{i=1}^{N} \left[\ell(y_i, \widehat{y}_i^{(m-1)}) + g_i f_m(\boldsymbol{x}_i) + \frac{1}{2} h_i f_m^2(\boldsymbol{x}_i) \right] + \gamma L + \frac{1}{2}\lambda \sum_{j=1}^{L} w_j^2, \tag{2.4}$$

where $g_i = \frac{\partial \ell(y_i, f)}{\partial f}\Big|_{f=\widehat{y}_i^{(m-1)}}$ and $h_i = \frac{\partial^2 \ell(y_i, f)}{\partial f^2}\Big|_{f=\widehat{y}_i^{(m-1)}}$ are the first and second derivatives of the loss function with respect to $\widehat{y}_i^{m-1}$. Ignoring the regularisation terms and the terms that does not depend on f_m, minimising (2.4) is equivalent to minimising

$$\begin{aligned} \widetilde{\mathcal{L}}^{(m)} &= \sum_{i=1}^{N} g_i f_m(\boldsymbol{x}_i) + \frac{1}{2} h_i f_m^2(\boldsymbol{x}_i) \\ &= \sum_{i=1}^{N} \frac{1}{2} h_i \left(f_m(\boldsymbol{x}_i) + \frac{g_i}{h_i} \right)^2 - \frac{1}{2} \frac{g_i^2}{h_i}. \end{aligned}$$

Hence, in a conceptual sense, the new tree f_m is fitted to targets $\{-g_i/h_i\}_{i=1}^N$ with weights h_i, which is why XGBoost is often described as a second-order gradient boosting method.

In practice, XGBoost does not explicitly fit a tree by regressing on $\{(\boldsymbol{x}_i, -g_i/h_i)\}_{i=1}^N$. Instead, it directly optimises the split structure and leaf weights using the aggregated gradients and Hessians. Let $q : \mathbb{R}^d \to \{1, 2, \dots, L\}$ denote the tree structure, which is a mapping from each input to a leaf index, and let $I_j = \{i : q(\boldsymbol{x}_i) = j\}$ be the set of training indices assigned to leaf j. For a fixed structure q, $\widetilde{\mathcal{L}}^{(m)}$ can be written as

$$\begin{aligned}
\widetilde{\mathcal{L}}^{(m)} &= \sum_{i=1}^{N} \left[g_i f_m(\boldsymbol{x}_i) + \frac{1}{2} h_i f_m^2(\boldsymbol{x}_i) \right] + \gamma L + \frac{1}{2} \lambda \sum_{j=1}^{L} w_j^2 \\
&= \sum_{j=1}^{L} \left[w_j \sum_{i \in I_j} g_i + \frac{1}{2} w_j^2 \left(\sum_{i \in I_j} h_i + \lambda \right) \right] + \gamma L.
\end{aligned}$$

The optimal weight for leaf j is therefore

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda},$$

and the substituting w_j^* back yields the optimal value of the objective for that fixed tree:

$$\widetilde{\mathcal{L}}^{(m)*} = -\frac{1}{2} \sum_{j=1}^{L} \frac{\left(\sum_{i \in I_j} g_i \right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma L.$$

To optimise the tree structure q, XGBoost typically starts from a single leaf and greedily adds splits. For a candidate split of a node with index set I into left and right children I_L and I_R, the loss reduction in $\widetilde{\mathcal{L}}^{(m)*}$ is

$$\begin{aligned}
\widetilde{\mathcal{L}}_{\text{split}} &= \widetilde{\mathcal{L}}^{(m)*}_{\text{before}} - \widetilde{\mathcal{L}}^{(m)*}_{\text{after}} \\
&= \frac{1}{2} \left[\frac{\left(\sum_{i \in I_L} g_i \right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i \right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{\left(\sum_{i \in I} g_i \right)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma.
\end{aligned}$$

A split is accepted if this gain is positive or exceeds a user-specified threshold, and the algorithm selects the split with the largest gain among the candidates, subject to any additional constraints such as maximum depth, minimum child weight, or minimum loss reduction.

2.3 Deep learning

2.3.1 Multilayer perceptron

A multilayer perceptron (MLP) is a class of feed-forward neural networks widely used for supervised learning tasks in modern deep learning. An MLP represents information using neurons. Each neuron stores a single scalar value, and a collection of neurons forms a layer, which represents a vector. An MLP typically consists of an input layer (which receives the input vector, such as features in tabular data or a vectorised image), several hidden layers (which learn intermediate representations), and an output layer (whose dimension and constraints depend on the task). For example, in a k-class classification problem, the output layer often has k neurons and produces values that can be interpreted as class probabilities. In a standard MLP, every neuron in one layer is connected to every neuron in the next layer through an affine transformation followed by a non-linear activation function. This is known as a fully connected layer, or dense layer. A sketch of an MLP is shown in Figure 2.1.

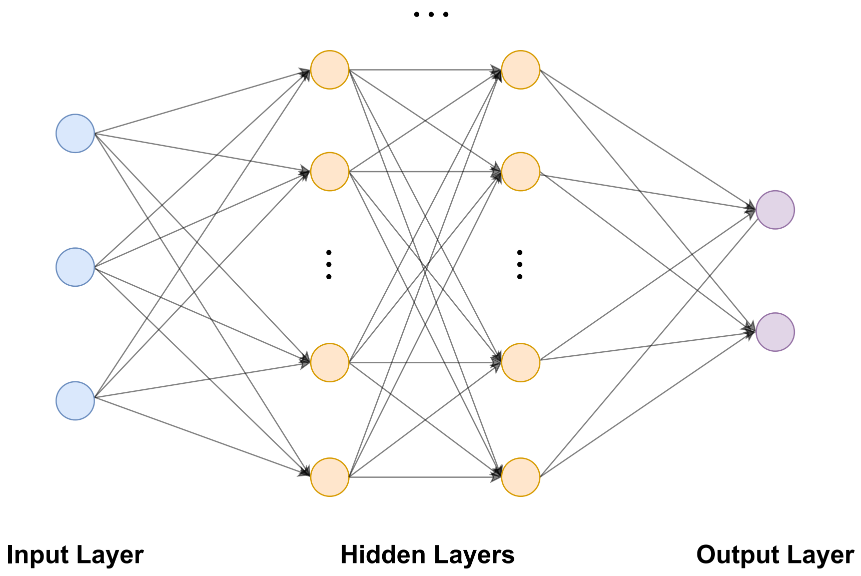

Figure 2.1: Illustrated architecture of an MLP with 3-dimensional input and 2-dimensional output. All neurons are fully connected to neurons in the next layers by an affine transformation followed by a nonlinear activation function.

Formally, consider a input vector $\boldsymbol{x} \in \mathbb{R}^{d_0}$ and an MLP with J layers. Let $\boldsymbol{W}^{(j)} \in \mathbb{R}^{d_j \times d_{j-1}}$ denote the learnable weight matrix between layer $j-1$ and j, and let $\boldsymbol{b}^{(j)} \in \mathbb{R}^{d_j}$ denote the learnable bias term in layer j. The forward pass through the network is given by

$$\begin{aligned} \boldsymbol{a}^{(0)} &= \boldsymbol{x} \\ \boldsymbol{a}^{(j)} &= \phi^{(j)}(\boldsymbol{W}^{(j)}\boldsymbol{a}^{(j-1)} + \boldsymbol{b}^{(j)}), \quad j = 1, \ldots, J, \end{aligned}$$

where $\phi^{(j)}$ is the element-wise nonlinear activation function (e.g., sigmoid, ReLU, tanh) in layer j. Equivalently, the MLP defines a nested composition of mappings:

$$f(\mathrm{x}; \boldsymbol{\theta}) = \phi^{(J)}\left(\mathbf{W}^{(J)}\phi^{(J-1)}\left(\cdots\phi^{(1)}\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right)\cdots\right) + \mathbf{b}^{(J)}\right),$$

where $\boldsymbol{\theta} = \{\boldsymbol{W}^{(j)}, \boldsymbol{b}^{(j)}\}_{j=1}^{J}$ is the all parameters.

To train the MLP over a dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$, we can choose a differentiable loss function $\ell(\cdot)$ that measures the discrepancy between the target y_i and the prediction $f(\boldsymbol{x}_i; \boldsymbol{\theta})$. Common choices include mean squared error for regression and cross-entropy loss for classification. Training typically minimises the regularised empirical risk

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \ell(f(\boldsymbol{x}_i; \boldsymbol{\theta}), y_i) + \lambda\Omega(\boldsymbol{\theta}); \quad \lambda > 0,$$

where $\Omega(\boldsymbol{\theta})$ is a regularisation term based on parameter norms and λ controls the strength of regularisation.

Parameters are commonly updated using gradient-based optimisation. The simplest form is gradient descent:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}\mathcal{L}, \tag{2.5}$$

where $\eta > 0$ is the learning rate. Computing $\nabla_{\boldsymbol{\theta}}\mathcal{L}$ requires applying the chain rule through all layers; the standard algorithm for doing this efficiently is known as backpropagation. In practice, variants such as stochastic gradient descent (Robbins and Monro, 1951) and adaptive methods like Adam (Kingma and Ba, 2015) are often used, but they rely on the same backpropagation gradients.

2.3.2 Long short-term memory

Long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) are a type of recurrent neural networks (RNN) widely used for modelling sequential data, such as time series. A defining feature of time series is temporal dependence: values at a given time point are often correlated with previous values. RNNs capture this dependence by maintaining a hidden state that is updated at each time step and carried forward through the sequence.

In a standard RNN, the hidden state $\boldsymbol{h}_t \in \mathbb{R}^h$ at time t depends on the current input $\boldsymbol{x}_t \in \mathbb{R}^d$ and the previous hidden state $\boldsymbol{h}_{t-1} \in \mathbb{R}^h$, thereby implicitly incorporating information from all past inputs up to time t. A common formulation is

$$\boldsymbol{h}_t = \phi(\boldsymbol{W}_h \boldsymbol{h}_{t-1} + \boldsymbol{W}_x \boldsymbol{x}_t + \boldsymbol{b}),$$

where ϕ is a non-linear activation function, typically $\tanh$, and $\boldsymbol{W}_h \in \mathbb{R}^{h\times h}$, $\boldsymbol{W}_x \in \mathbb{R}^{h\times d}$, and $\boldsymbol{b} \in \mathbb{R}^h$ are learnable parameters. While RNNs are, in principle, able to represent long-term dependencies, they often struggle in practice because gradients can vanish or explode during training with backpropagation through time (BPTT).

To illustrate this issue in a simplified univariate setting, consider a loss function ℓ and the gradient with respect to the hidden state at an earlier time step k. Using BPTT,

$$\frac{\partial \ell}{\partial h_k} = \frac{\partial \ell}{\partial h_T} \prod_{t=k+1}^{T} \frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial \ell}{\partial h_T} \cdot W_h^{T-k} \prod_{t=k+1}^{T} \phi'(W_h h_{t-1} + W_x x_t + b). \tag{2.6}$$

For common activation functions such as tanh, $|\phi'(\cdot)| < 1$, and it is often strictly less than 1 over much of their range. As a result, the gradient magnitude is governed by repeated multiplication of terms, and is strongly influenced by the scale of W_h^{T-k}. Consequently, gradients tend to vanish when $|W_h| < 1$ and can explode when $|W_h| > 1$, making it difficult to learn dependencies over long time horizons.

LSTM networks alleviate this problem by introducing an explicit memory cell with an additive update, together with gating mechanisms that control information flow. An LSTM unit maintains a cell state $\boldsymbol{c}_t \in \mathbb{R}^h$ to carry long-term memory, and uses three gates: an input gate $\boldsymbol{i}_t$, an output gate $\boldsymbol{o}_t$, and a forget gate $\boldsymbol{f}_t$. These gates determine what information is written to memory, what is retained from the past, and what is exposed as the hidden state. Given an input vector $\boldsymbol{x}_t \in \mathbb{R}^d$ and a previous hidden state $\boldsymbol{h}_{t-1} \in \mathbb{R}^h$, the gates are computed as:

$$\begin{aligned}
\boldsymbol{i}_t &= \text{sigmoid}(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i), \\
\boldsymbol{o}_t &= \text{sigmoid}(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o), \\
\boldsymbol{f}_t &= \text{sigmoid}(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f),
\end{aligned}$$

where $\operatorname{sigmoid}(\cdot)$ is the sigmoid activation function, and $\boldsymbol{W}_{(\cdot)} \in \mathbb{R}^{h \times d}$, $\boldsymbol{U}_{(\cdot)} \in \mathbb{R}^{h \times h}$, and $\boldsymbol{b}_{(\cdot)} \in \mathbb{R}^{h}$ are the learnable parameters. Because the sigmoid maps to $(0, 1)$, each gate can be interpreted as a soft switch that controls how much information passes through.

With the gates defined, the cell state $\boldsymbol{c}_t \in \mathbb{R}^h$ and hidden state $\boldsymbol{h}_t \in \mathbb{R}^h$ are then updated via:

$$\begin{aligned}
\widetilde{\boldsymbol{c}}_t &= \tanh(\boldsymbol{W}_c \boldsymbol{x}_t + \boldsymbol{U}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c), \\
\boldsymbol{c}_t &= \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \widetilde{\boldsymbol{c}}_t, \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t),
\end{aligned}$$

where $\tanh(\cdot)$ is the hyperbolic tangent function, $\odot$ denotes element-wise Hadamard product, and $\boldsymbol{W}_c$, $\boldsymbol{U}_c$, and $\boldsymbol{b}_c$ are additional learnable parameters. The intermediate quantity $\widetilde{\boldsymbol{c}}_t$ is the candidate memory content computed from the current input and previous hidden state. The overall architecture is illustrated in Figure 2.2. The additive form of the cell-state update helps stabilise gradient propagation over time, avoiding the exploding-gradient behaviours often observed in standard RNNs. To see this in the same univariate setting as in (2.6), apply BPTT through the cell states. Since $c_t = f_t c_{t-1} + i_t \widetilde{c}_t$, we have $\frac{\partial c_t}{\partial c_{t-1}} = f_t$, and

$$\frac{\partial \ell}{\partial c_k} = \frac{\partial \ell}{\partial c_T} \prod_{t=k+1}^{T} \frac{\partial c_t}{\partial h_{c-1}} = \frac{\partial \ell}{\partial c_T} \prod_{t=k+1}^{T} f_t.$$

Because each $f_t \in (0, 1)$, the product $\prod_{t=k+1}^{T} f_t$ cannot blow up. This rules out the kind of gradient explosion that can occur in a standard RNN, where repeated multiplication by an unconstrained recurrent weight can lead to exponential growth.

The hidden state $\boldsymbol{h}_t$ is typically treated as the output representation at time t. It can be mapped to a prediction as a one-step-ahead forecast via an additional output layer, such as a linear projection whose output dimension matches the target variable. Model parameters in LSTM can be updated by minimising a suitable loss function using gradient-based optimisation.

2.3.3 Normalising flows

Normalising flows are a class of generative models that represent a probability distribution by applying a sequence of invertible and differentiable transformations to a simple base distribution. Each transformation is differentiable, and so is its inverse; such a map is called a diffeomorphism. Let $\boldsymbol{Z} \in \mathbb{R}^d$ be a random vector with density $p_{\boldsymbol{Z}}(\boldsymbol{z})$, and let $T : \mathbb{R}^d \to \mathbb{R}^d$ be a diffeomorphism.

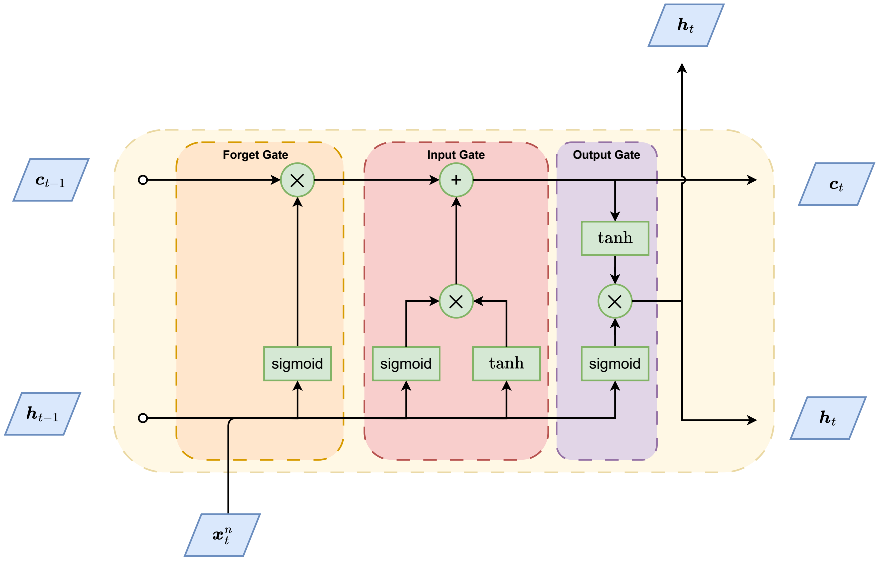

Figure 2.2: Architecture of an LSTM block that updates the cell state $\boldsymbol{c}_{t-1}$ and hidden state $\boldsymbol{h}_{t-1}$ using the current input $\boldsymbol{x}_t$. The circular + denotes element-wise addition, while the circular × denotes element-wise multiplication. Rectangles indicate activation functions, labelled accordingly.

Define $\boldsymbol{X} = T(\boldsymbol{Z})$, then the density of $\boldsymbol{X}$ is given by the change-of-variables formula

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = p_{\boldsymbol{Z}}(\boldsymbol{z}) \left|\det J_T(\boldsymbol{z})\right|^{-1}, \tag{2.7}$$

where $J_T(\boldsymbol{z})$ is the $d \times d$ Jacobian matrix of T evaluated at $\boldsymbol{z}$, and $\boldsymbol{x} = T(\boldsymbol{z})$. Although (2.7) is just the change-of-variables formula, one can show that for any well-behaved densities $p_{\boldsymbol{Z}}$ and $p_{\boldsymbol{X}}$, there exists an invertible transformation T that maps $\boldsymbol{Z}$ to $\boldsymbol{X}$ (Papamakarios et al., 2021).

One way to see this is via an autoregressive factorisation. We can write

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{j=1}^{d} p(x_j \mid \boldsymbol{x}_{<j}),$$

and define a mapping $F = (F_1, \ldots, F_d)$ elementwise by

$$u_j = F_j(x_j; \boldsymbol{x}_{<j}) = \int_{-\infty}^{x_j} p(t \mid \boldsymbol{x}_{<j}) \mathrm{d}t.$$

For each j, $F_j(\cdot; \boldsymbol{x}_{<j})$ is the conditional cumulative distribution function (CDF) of $X_j \mid \boldsymbol{X}_{<j}$. It is non-decreasing and satisfies

$$\frac{\partial F_j}{\partial x_j} = p(x_j \mid \boldsymbol{x}_{<j}).$$

Under mild assumptions, F is invertible and differentiable almost everywhere. Note that F_j

depends only on $(x_1, \dots, x_j)$. Therefore, its Jacobian is lower triangular and

$$\det J_F(\boldsymbol{x}) = \det \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \frac{\partial F_d}{\partial x_1} & \cdots & \frac{\partial F_d}{\partial x_d} \end{bmatrix} = \prod_{j=1}^{d} \frac{\partial F_j}{\partial x_j} = \prod_{j=1}^{d} p(x_j \mid \boldsymbol{x}_{<j}) = p_{\boldsymbol{X}}(\boldsymbol{x}).$$

Applying the change-of-variables formula yields

$$p_{\boldsymbol{U}}(\boldsymbol{u}) = p_{\boldsymbol{X}}(\boldsymbol{x}) \left|\det J_F(\boldsymbol{x})\right|^{-1} = p_{\boldsymbol{X}}(\boldsymbol{x}) \left|p_{\boldsymbol{X}}(\boldsymbol{x})\right|^{-1} = 1, \tag{2.8}$$

so $\boldsymbol{U}$ is uniform on $[0, 1]^d$. In the univariate case, this is known as the probability integral transform. Similarly, we can construct a transformation G that maps $\boldsymbol{Z}$ to a uniform random vector on $[0, 1]^d$. Composing these maps gives $T = F^{-1} \circ G$, which maps $\boldsymbol{Z}$ to $\boldsymbol{X}$.

The argument above establishes existence in principle. In practice, normalising flows approximate such a transformation by composing several learnable diffeomorphisms. Given K learnable diffeomorphisms $\mathcal{T}_k$, $k = 1, \dots, K$, we define

$$\boldsymbol{X} = (\mathcal{T}_K \circ \cdots \circ \mathcal{T}_1)(\boldsymbol{Z}),$$

and the resulting density is

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = p_{\boldsymbol{Z}}(\boldsymbol{z}) \left|\det J_{\mathcal{T}_K \circ \cdots \circ \mathcal{T}_1}(\boldsymbol{z})\right|^{-1} = p_{\boldsymbol{Z}}(\boldsymbol{z}) \prod_{k=1}^{K} \left|\det J_{\mathcal{T}_k}(\boldsymbol{z}^{(k)})\right|^{-1},$$

where $\boldsymbol{z}^{(1)} = \boldsymbol{z}$ and $\boldsymbol{z}^{(k+1)} = \mathcal{T}_k(\boldsymbol{z}^{(k)})$. The learnable $\mathcal{T}_k$ is often parameterised using neural networks to improve expressivity, while the architecture is designed so that invertibility and tractable Jacobian determinants are preserved. Common families include autoregressive flows, linear flows, and residual flows. Below, we describe Real NVP (Dinh et al., 2017), a widely used coupling-based autoregressive flow with affine transformations.

Real NVP. In a Real NVP layer $\mathcal{T}_k$, the input $\boldsymbol{z}^{(k)} \in \mathbb{R}^d$ is split into two blocks, $\boldsymbol{z}_{b_1}^{(k)}$ and $\boldsymbol{z}_{b_2}^{(k)}$, where $b_1 \cup b_2 = \{1, \dots, d\}$ and $b_1 \cap b_2 = \varnothing$. Often $|b_1| = |b_2|$, and the roles of b_1 and b_2 are alternated across layers or mixed via permutations so that all dimensions are eventually transformed.

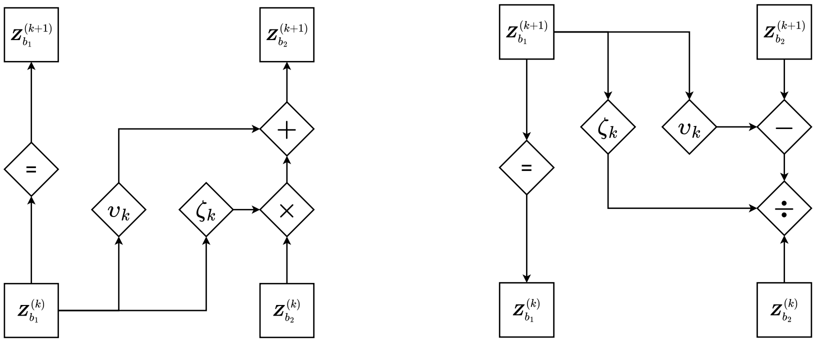

Figure 2.3: Forward (left) and backward (right) propagation in a layer of a Real NVP model. The input vector $\boldsymbol{z}$ is partitioned into two subvectors $\boldsymbol{z}_{b_1}$ and $\boldsymbol{z}_{b_2}$ of the same dimension. ζ_k and υ_k are two neural networks in layer k that take $\boldsymbol{z}_{b_1}$ as input and output vectors used as the scale and translation in an affine transformation, respectively.

The coupling layer leaves $\boldsymbol{z}_{b_1}^{(k)}$ unchanged and applies an affine transformation to $\boldsymbol{z}_{b_2}^{(k)}$:

$$\begin{aligned}\boldsymbol{z}_{b_1}^{(k+1)} &= \boldsymbol{z}_{b_1}^{(k)},\\ \boldsymbol{z}_{b_2}^{(k+1)} &= \exp\{\zeta_k(\boldsymbol{z}_{b_1}^{(k)})\} \odot \boldsymbol{z}_{b_2}^{(k)} + \upsilon_k(\boldsymbol{z}_{b_1}^{(k)}).\end{aligned}$$

Here, ζ_k and υ_k are neural networks (e.g. MLPs) that output vectors of the same dimension as $\boldsymbol{z}_{b_2}^{(k)}$. Intuitively, ζ_k produces a log-scale and υ_k produces a shift, both conditioned on $\boldsymbol{z}_{b_1}^{(k)}$, allowing the model to capture dependencies between the two blocks.

A key advantage is that the Jacobian of $\mathcal{T}_k$ is triangular, so its determinant is easy to compute:

$$\det J_{\mathcal{T}_k}(\boldsymbol{z}^{(k)}) = \prod_{j \in b_2} \exp\{\zeta_k(\boldsymbol{z}_{b_1}^{(k)})_j\}.$$

Thus, the log-determinant depends only on the output of ζ_k, not on the internal complexity of the networks, which keeps likelihood evaluation tractable.

Training a normalising flow is similar to fitting a parametric statistical model: because the likelihood can be evaluated explicitly, the parameters of the transformations (i.e., the neural network weights) can be learned by minimising the negative log-likelihood using gradient-based optimisation.

2.3.4 Diffusion models

Diffusion models, in particular Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020), are a class of generative models that have achieved strong performance in computer vision tasks such as image and video generation. Given data $\boldsymbol{x}^{(0)} \sim q(\boldsymbol{x}^{(0)})$, the goal of diffusion models is to learn a model distribution $p_\theta(\boldsymbol{x}^{(0)})$ that approximates the unknown data distribution and from which we can sample efficiently. Diffusion models learn $p_\theta(\boldsymbol{x}^{(0)})$ through two coupled processes inspired by nonequilibrium thermodynamics (Sohl-Dickstein et al., 2015):

1. a forward (diffusion) process that gradually adds Gaussian noise to the data, and

2. a reverse process that learns to remove the noise and recover a clean sample.

Forward (diffusion) process. In the forward process, the data are transformed over T discrete time steps into progressively noisier latent variables $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(T)}$ via the Markov transitions.

$$q(\boldsymbol{x}^{(t)} \mid \boldsymbol{x}^{(t-1)}) := \mathcal{N}(\boldsymbol{x}^{(t)}; \sqrt{1-\beta_t}\boldsymbol{x}^{(t-1)}, \beta_t \boldsymbol{I}), \tag{2.9}$$

where $\{\beta_t\}_{t=1}^T$ is a fixed noise schedule and $\boldsymbol{I}$ is the identity matrix. Define $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$. It follows that the marginal distribution of $\boldsymbol{x}^{(t)}$ given the original data $\boldsymbol{x}^{(0)}$ has the closed form

$$q(\boldsymbol{x}^{(t)} \mid \boldsymbol{x}^{(0)}) = \mathcal{N}(\boldsymbol{x}^{(t)}; \sqrt{\bar{\alpha}_t}\boldsymbol{x}^{(0)}, (1-\bar{\alpha}_t)\boldsymbol{I}), \tag{2.10}$$

which implies that, for sufficiently large T and an appropriate noise schedule, $\boldsymbol{x}^{(T)}$ is close in distribution to $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, largely independent of the original distribution $q(\boldsymbol{x}^{(0)})$. The full forward process can be written as

$$q(\boldsymbol{x}^{(1:T)} | \boldsymbol{x}^{(0)}) = \prod_{n=1}^{T} q(\boldsymbol{x}^{(t)} \mid \boldsymbol{x}^{(t-1)}).$$

Reverse (denoising) process. The reverse process aims to invert the diffusion by sequentially denoising. Starting from a simple base distribution $p(\boldsymbol{x}^{(T)}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, DDPMs define a reverse-time Markov chain with Gaussian transitions:

$$p_\theta(\boldsymbol{x}^{(t-1)} \mid \boldsymbol{x}^{(t)}) = \mathcal{N}(\boldsymbol{x}^{(t-1)}; \boldsymbol{\mu}_\theta(\boldsymbol{x}^{(t)}, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}^{(t)}, t)),$$
$$p_\theta(\boldsymbol{x}^{(0:T)}) = p(\boldsymbol{x}^{(T)}) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}^{(t-1)} \mid \boldsymbol{x}^{(t)}).$$

The functions $\boldsymbol{\mu}_\theta(\boldsymbol{x}^{(t)}, t)$ and $\boldsymbol{\Sigma}_\theta(\boldsymbol{x}^{(t)}, t)$ are learned so that the reverse chain produces samples $\boldsymbol{x}^{(0)}$ resembling the data distribution.

Training via variational inference. Training a diffusion model is commonly formulated as a variational inference problem with $\boldsymbol{x}^{(1:T)}$ treated as latent variables. One can decompose the log-likelihood as

$$\begin{aligned}\log p_\theta(\boldsymbol{x}^{(0)}) &= \mathrm{KLD}\{q(\boldsymbol{x}^{(1:T)} \mid \boldsymbol{x}^{(0)}) \parallel p_\theta(\boldsymbol{x}^{(1:T)} \mid \boldsymbol{x}^{(0)})\} + \mathbb{E}_q\left[\log \frac{p_\theta(\boldsymbol{x}^{(0:T)})}{q(\boldsymbol{x}^{(1:T)}|\boldsymbol{x}^{(0)})}\right] \\ &\geq \mathbb{E}_q\left[\log \frac{p_\theta(\boldsymbol{x}^{(0:T)})}{q(\boldsymbol{x}^{(1:T)}|\boldsymbol{x}^{(0)})}\right] := \mathrm{ELBO},\end{aligned}$$

where ELBO denotes the evidence lower bound. Maximising the ELBO provides a tractable objective for learning $\boldsymbol{\theta}$. Since the forward process $q(\boldsymbol{x}^{(1:T)}|\boldsymbol{x}^{(0)})$ is fixed and known, improving the ELBO encourages the reverse process $p_\theta(\boldsymbol{x}^{(0:T)})$ to match the implied reverse dynamics of the diffusion, thereby increasing $\log p_\theta(\boldsymbol{x}^{(0)})$.

Ho et al. (2020) shows that the ELBO can be decomposed into a sum of KL terms across timesteps:

$$\begin{aligned}&\mathbb{E}_q\left[\log \frac{p_\theta(\boldsymbol{x}^{(0:T)})}{q(\boldsymbol{x}^{(1:T)}|\boldsymbol{x}^{(0)})}\right] \\ =&\mathbb{E}_q\left[\log p(\boldsymbol{x}^{(T)}) + \sum_{n=2}^{T} \log \frac{p_\theta(\boldsymbol{x}^{(t-1)} \mid \boldsymbol{x}^{(t)})}{q(\boldsymbol{x}^{(t)}|\boldsymbol{x}^{(t-1)}, \boldsymbol{x}^{(0)})} + \log \frac{p_\theta(\boldsymbol{x}^{(0)} \mid \boldsymbol{x}^{(1)})}{q(\boldsymbol{x}^{(1)}|\boldsymbol{x}^{(0)})}\right] \\ =&\mathbb{E}_q\left[\log p(\boldsymbol{x}^{(T)}) + \sum_{n=2}^{T} \log \left\{\frac{p_\theta(\boldsymbol{x}^{(t-1)} \mid \boldsymbol{x}^{(t)})}{q(\boldsymbol{x}^{(t-1)}|\boldsymbol{x}^{(t)}, \boldsymbol{x}^{(0)})} \frac{q(\boldsymbol{x}^{(t-1)} \mid \boldsymbol{x}^{(0)})}{q(\boldsymbol{x}^{(t)} \mid \boldsymbol{x}^{(0)})}\right\} + \log \frac{p_\theta(\boldsymbol{x}^{(0)} \mid \boldsymbol{x}^{(1)})}{q(\boldsymbol{x}^{(1)}|\boldsymbol{x}^{(0)})}\right] \\ =&\mathbb{E}_q\left[\log \frac{p(\boldsymbol{x}^{(T)})}{q(\boldsymbol{x}^{(T)} \mid \boldsymbol{x}^{(0)})} + \sum_{n=2}^{T} \log \frac{p_\theta(\boldsymbol{x}^{(t-1)} \mid \boldsymbol{x}^{(t)})}{q(\boldsymbol{x}^{(t-1)} \mid \boldsymbol{x}^{(t)}, \boldsymbol{x}^{(0)})} + \log p_\theta(\boldsymbol{x}^{(0)} \mid \boldsymbol{x}^{(1)})\right] \\ =&\mathbb{E}_q\left[-\underbrace{\mathrm{KLD}\{q(\boldsymbol{x}^{(T)} \mid \boldsymbol{x}^{(0)}) \parallel p(\boldsymbol{x}^{(T)})\}}_{L_T} - \sum_{n=2}^{T} \underbrace{\mathrm{KLD}\{q(\boldsymbol{x}^{(t-1)} \mid \boldsymbol{x}^{(t)}, \boldsymbol{x}^{(0)}) \parallel p_\theta(\boldsymbol{x}^{(t-1)} \mid \boldsymbol{x}^{(t)})\}}_{L_{t-1}} + \underbrace{\log p_\theta(\boldsymbol{x}^{(0)} \mid \boldsymbol{x}^{(1)})}_{L_0}\right].\end{aligned}$$

The first term L_T is constant during training because both $q(\boldsymbol{x}^{(T)} \mid \boldsymbol{x}^{(0)})$ and $p(\boldsymbol{x}^{(T)})$ are known. The remaining terms involve matching the learned reverse transition $p_\theta(\boldsymbol{x}^{(t-1)} \mid \boldsymbol{x}^{(t)})$ to the true reverse posterior $q(\boldsymbol{x}^{(t-1)} \mid \boldsymbol{x}^{(t)}, \boldsymbol{x}^{(0)})$, which is Gaussian and available in closed form:

$$\begin{aligned}q(\boldsymbol{x}^{(t-1)} \mid \boldsymbol{x}^{(t)}, \boldsymbol{x}^{(0)}) &= \mathcal{N}(\boldsymbol{x}^{(t-1)}; \widetilde{\mu}_t(\boldsymbol{x}^{(t)}, \boldsymbol{x}^{(0)}), \widetilde{\beta}_t \boldsymbol{I}), \\ \widetilde{\mu}_t(\boldsymbol{x}^{(t)}, \boldsymbol{x}^{(0)}) &= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}^{(0)} + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}^{(t)}, \\ \widetilde{\beta}_t &= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t,\end{aligned}$$

Assuming $\Sigma_\theta(\boldsymbol{x}^{(t)}, t) = \sigma_t^2 \boldsymbol{I}$, with σ_t^2 set to either β_t or $\widetilde{\beta}_t$, the KL term L_{t-1} reduces to a weighted squared-error objective:

$$\mathbb{E}_q(L_{t-1}) = \mathbb{E}_q \left[\frac{\left\| \boldsymbol{\mu}_\theta(\boldsymbol{x}^{(t)}, t) - \widetilde{\boldsymbol{\mu}}_t(\boldsymbol{x}^{(t)}, \boldsymbol{x}^{(0)}) \right\|^2}{2\sigma_t^2} \right] + C, \tag{2.11}$$

where C is constant with respect to θ. Thus, training encourages $\boldsymbol{\mu}_\theta(\boldsymbol{x}^{(t)}, t)$ to match the true posterior mean $\widetilde{\boldsymbol{\mu}}_t(\boldsymbol{x}^{(t)}, \boldsymbol{x}^{(0)})$.

Using (2.10), we can reparameterise $\boldsymbol{x}^{(t)}$ as

$$\boldsymbol{x}^{(t)} = \sqrt{\bar{\alpha}_t}\boldsymbol{x}^{(0)} + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}),$$

or equivalently,

$$\boldsymbol{x}^{(0)} = \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}^{(t)} - \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}).$$

Substituting this into $\widetilde{\boldsymbol{\mu}}_t(\boldsymbol{x}^{(t)}, \boldsymbol{x}^{(0)})$ yields an equivalent parametrisation in terms of the noise:

$$\widetilde{\boldsymbol{\mu}}_t(\boldsymbol{x}^{(t)}, \boldsymbol{\epsilon}) = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}^{(t)} - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon} \right), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{2.12}$$

This shows that learning $\boldsymbol{\mu}_\theta(\boldsymbol{x}^{(t)}, t)$ is closely related to learning the noise term $\boldsymbol{\epsilon}$ at each timestep. Ho et al. (2020) therefore proposed parameterising the reverse mean via a neural network $\boldsymbol{\epsilon}_\theta$ that predicts the noise:

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}^{(t)}, t) = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}^{(t)} - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\boldsymbol{x}^{(t)}, t) \right).$$

With this parameterisation, (2.11) simplifies to the weighted noise-prediction loss

$$\mathbb{E}_q(L_{t-1}) - C = \mathbb{E}_{\mathbf{x}^{(0)}, \boldsymbol{\epsilon}} \left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1 - \bar{\alpha}_t)} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(\sqrt{\bar{\alpha}_t}\mathbf{x}^{(0)} + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t \right) \right\|^2 \right]. \tag{2.13}$$

The final term L_0 corresponds to the likelihood model $p_\theta(\boldsymbol{x}^{(0)} \mid \boldsymbol{x}^{(1)})$. Maximising L_0 with respect to $\boldsymbol{\mu}_\theta$ yields an objective of the same general form, and under the Gaussian assumption, it is optimised when $\boldsymbol{\mu}_\theta(\boldsymbol{x}^{(1)}, 1)$ matches $\boldsymbol{x}^{(0)}$. This leads to a squared-error loss consistent with (2.13):

$$\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(\sqrt{\bar{\alpha}_1}\mathbf{x}^{(0)} + \sqrt{1 - \bar{\alpha}_1}\boldsymbol{\epsilon}, 1 \right) \right\|^2.$$

Finally, Ho et al. (2020) show that it is effective in practice to minimise a simplified version of the (negative) ELBO:

$$\mathbb{E}_{t,\mathbf{x}^{(0)},\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}^{(0)}+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},t\right)\right\|^2\right] \tag{2.14}$$

which is straightforward to implement and often improves sample quality.

After training, sampling proceeds by drawing $\boldsymbol{x}^{(T)} \sim \mathcal{N}(\mathbf{0},\boldsymbol{I})$ and then iterating the reverse updates

$$\boldsymbol{x}^{(t-1)}=\frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}^{(t)}-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}^{(t)},t)\right)+\sigma_t\boldsymbol{z},\quad t=T,\dots,1,$$

where $\boldsymbol{z}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})$ for $t>1$, and $\boldsymbol{z}=\mathbf{0}$ for $t=1$.

Chapter 3

A Bayesian multivariate extreme value mixture model

Accurate risk assessment for natural hazards requires models that describe both extreme and non-extreme events. In multivariate settings, extreme value theory (EVT) provides asymptotically justified models for exceedances over high thresholds via the multivariate generalised Pareto distribution (mGPD), but does not capture the bulk of the distribution. In this Chapter, we propose a Bayesian multivariate extreme mixture model that jointly models the bulk and tail regions while preserving the exact tail properties guaranteed by EVT. The bulk is modelled parametrically and the tail with the mGPD, with both components defined on disjoint regions separated by a multivariate threshold vector considered an unknown parameter. This formulation decouples the bulk and tail, preserving theoretical tail properties under changes to the bulk, while naturally incorporating uncertainty quantification for all parameters, including thresholds. Posterior inference is performed using the automated factor slice sampler, which efficiently explores the strong dependencies between parameters. Simulation studies show that the model recovers parameters accurately across a range of tail behaviours, remains robust under model misspecification, and estimates extremal dependence measures reliably. An application to UK temperature extremes shows that the model captures the moderate and high quantiles accurately.

3.1 Introduction

In many applications, risk depends on both extreme and non-extreme events. For example, flooding may arise from intense short-term precipitation or from sustained moderate rainfall over several days. Classical extreme value theory (EVT) provides asymptotically justified models for the tail of a distribution, but does not describe the bulk, and is therefore insufficient when the full range of the data is of interest.

In the univariate setting, this limitation has motivated numerous approaches that combine a generalised Pareto distribution (GPD) for threshold exceedances with a lighter-tailed distribution for the remainder of the data. Examples include the dynamically weighted mixtures of Frigessi et al. (2002) and the Bayesian threshold models of Behrens et al. (2004), as well as semi- and non-parametric extensions by Tancredi et al. (2006); do Nascimento et al. (2012); MacDonald et al. (2011) and Huang et al. (2019). Other unified approaches, such as Naveau et al. (2016) and Stein (2021), reproduce GPD behaviour in both tails without the need for explicit thresholding. Krock et al. (2022) extended Stein's model to accommodate nonstationary settings, while Opitz et al. (2018a) and Castro-Camilo et al. (2019) employed discontinuous spliced models to represent the bulk and tail of environmental variables over space and time, leading to improved prediction and forecasting of extreme events.

In the multivariate case, both marginal distributions and dependence structures must be modelled. A common approach transforms margins to a standard scale and models dependence via copulas, often in a mixture form to represent both bulk and tail behaviour (Aulbach et al., 2012b; André et al., 2024). When margins are also of interest, models combining marginal mixture approaches with copula-based dependence have been proposed (Leonelli and Gamerman, 2020). However, existing copula-based multivariate mixture frameworks suffer from entanglement between the bulk and tail: the bulk copula can influence the tail region, and the tail copula remains sensitive to bulk data, even with optimised weighting. This dependency could become problematic in small samples, where limited tail observations lead to fragile tail inference. Piecewise copula constructions, such as that of Aulbach et al. (2012a), offer partial relief but cannot simultaneously model the margins.

To address these issues, we develop a Bayesian multivariate extreme mixture model that combines a flexible parametric bulk distribution with the multivariate generalised Pareto distribution (mGPD) for the tail, defined on disjoint regions separated by a multivariate threshold vector. The threshold vector is treated as an unknown parameter, allowing the data to determine its value and enabling principled uncertainty quantification. Bulk and tail dependence structures are specified independently, ensuring that the exact tail properties of the mGPD are preserved regardless of the bulk choice. Posterior inference is carried out using the automated factor slice sampler (Tibbits et al., 2014), which efficiently explores strong dependencies between bulk, tail, and threshold parameters in the posterior distribution. Through simulations, we demonstrate that our framework recovers parameters accurately under a range of tail behaviours, remains robust to model misspecification, and estimates extremal dependence measures reliably. Our method's applicability is illustrated by analysing UK temperature records containing extreme events.

The remainder of this paper is organised as follows. Section 3.2 presents our model, including a detailed overview of the mGPD and our chosen representation. Section 3.3 describes the prior specifications and the methodology for posterior inference. Simulation studies are presented in Section 3.4, and Section 3.5 applies the model to UK temperature data. Section 3.6 discusses practical considerations for implementation, and Section 3.7 concludes.

3.2 Multivariate extreme mixture model

3.2.1 Multivariate generalised Pareto distribution

Let $\boldsymbol{Y}_1, \dots \boldsymbol{Y}_n$ be n independent and identical copies of the d-dimensional random vector $\boldsymbol{Y}$, which is in the max domain of attraction of an mGEVD G. This means that there exists sequences of normalising vectors $\boldsymbol{\alpha}_n \in (0, \infty)^d$ and $\boldsymbol{\beta}_n \in \mathbb{R}^d$ such that

$$\mathbb{P}\left(\boldsymbol{\alpha}_n \{\max_{1\le i\le n} \boldsymbol{Y}_i\} + \boldsymbol{\beta}_n \le \boldsymbol{y}\right) = \mathbb{P}^n\left(\boldsymbol{\alpha}_n \boldsymbol{Y} + \boldsymbol{\beta}_n \le \boldsymbol{y}\right) \to G(\boldsymbol{y}), \quad n \to \infty. \tag{3.1}$$

Here, operations on vectors are componentwise. For example, $\boldsymbol{\alpha}_n \boldsymbol{Y} + \boldsymbol{\beta}_n = (\alpha_{n,1} Y_1 + \beta_{n,1}, \cdots, \alpha_{n,d} Y_d + \beta_{n,d})$ and $\max_{1\le i\le n}\{\boldsymbol{Y}_i\} = (\max_{1\le i\le n}\{Y_{i,1}\}, \cdots, \max_{1\le i\le n}\{Y_{i,d}\})$. $G(\boldsymbol{y})$ is an mGEVD with non-degenerating margins $G_j(x)$, $j = 1, \dots, d$ belonging to the univariate generalised extreme value family of distributions. Without loss of generality, we can assume that marginals are unit Fréchet with cumulative distribution function $\exp(-y^{-1})$, for $y > 0$. In that case, we can write the joint distribution function G as

$$G(\boldsymbol{y}) = \exp\left\{-\int_{\mathcal{S}_d} \max\left(\frac{\omega_1}{y_1}, \dots, \frac{\omega_d}{y_d}\right) \mathrm{d}Q(\boldsymbol{\omega})\right\}, \quad \boldsymbol{y} > \boldsymbol{0}. \tag{3.2}$$

Q is called the spectral measure, an arbitrary positive finite measure over the unit simplex $\mathcal{S}_d = \{\boldsymbol{\omega} \in [0,1]^d : \sum_{j=1}^d \omega_j = 1\}$ that satisfies the constraint

$$\int_{\mathcal{S}_d} w_j \mathrm{d}Q(\boldsymbol{w}) = 1, \quad j = 1, \dots, d.$$

Let $\boldsymbol{Y} \nleq \boldsymbol{\beta}_n$ denote the event of at least one of the $\boldsymbol{Y}$ components exceeding the corresponding $\boldsymbol{\beta}_n$ component. If the convergence in (3.1) holds, the conditional random vector defined as

$$\frac{\boldsymbol{Y}-\boldsymbol{\beta}_n}{\boldsymbol{\alpha}_n}\bigg|\boldsymbol{Y} \nleq \boldsymbol{\beta}_n \tag{3.3}$$

converges in distribution to a d-dimensional random vector $\boldsymbol{X}$ with multivariate generalised Pareto distribution (mGPD) H (Rootzén and Tajvidi, 2006). Based on the relationship in (3.3), H can be expressed as in terms of G by

$$H(\boldsymbol{x}) = \frac{1}{\log G(\boldsymbol{0})} \log \frac{G(\boldsymbol{x} \wedge \boldsymbol{0})}{G(\boldsymbol{x})}, \tag{3.4}$$

where $\wedge$ denotes the componentwise minimum and $0 < G(\boldsymbol{0}) < 1$ is assumed. The marginal distributions of $H(\boldsymbol{x})$ do not conform to univariate GPDs since $\boldsymbol{Y} \nleq \boldsymbol{\beta}_n$ does not necessarily imply that $Y_j > \beta_j$ for every j. However, if we condition on $X_j > 0$, then the conditional margin $H_j(x_j|x_j > 0)$ are univariate GPD, i.e.,

$$\mathbb{P}(X_j > x_j|X_j > 0) = 1 - \frac{H_j(x) - H_j(0)}{1 - H_j(0)} = \left(1 + \gamma_j \frac{x_j}{\sigma_j}\right)^{-1/\gamma_j}, \quad x_j > 0. \tag{3.5}$$

The lower endpoint η_j^l and upper endpoint η_j^u of H_j are $\eta_j^l = -\sigma_j/\gamma_j$ and $\eta_j^u = +\infty$ for $\gamma_j > 0$, $\eta_j^l = -\infty$ and $\eta_j^u = -\sigma_j/\gamma_j$ for $\gamma_j < 0$, and $\eta_j^l = -\infty$ and $\eta_j^u = +\infty$ for $\gamma_j = 0$. As we can see from (3.2) and (3.4), both the mGEVD and the mGPD depend on the spectral measure Q and therefore have no unique representation. Based on the work by Rootzén et al. (2018b), Kiriliouk et al. (2019) propose three equivalent representations of the mGPD density, denoted R, U, and T, for use in parametric modelling. These forms can be transformed into one another via a suitable change of variable. The choice of representation depends on the intended application (see Kiriliouk et al. (2019) for details), and does not affect our framework. Here, we use the U representation with density

$$h_{\boldsymbol{U}}(\boldsymbol{x}) = \mathbb{1}\{\max(\boldsymbol{x}) > 0\} \frac{\prod_{j=1}^d (\gamma_j x_j + \sigma_j)^{-1}}{\mathbb{E}[\exp(\max(\boldsymbol{U}))]} \int_0^\infty f_{\boldsymbol{U}}\left(\frac{1}{\boldsymbol{\gamma}} \log\left(\frac{\boldsymbol{\gamma}}{\boldsymbol{\sigma}}\boldsymbol{x} + \boldsymbol{1}\right) + \log t\right) \mathrm{d}t, \tag{3.6}$$

where $f_{\boldsymbol{U}}$ is called a generator and is the density of a random vector $\boldsymbol{U}$ satisfying $\mathbb{E}[\exp(U_j)] < +\infty, j = 1, \cdots, d$, and $[\log(\gamma_j x_j/\sigma_j + 1)]/\gamma_j$ takes its liming form x_j/σ_j when $\gamma_j = 0$. A standardized form of (3.6) with $\boldsymbol{\gamma} = 0$ and $\boldsymbol{\sigma} = 1$ can be obtained by applying the transformation

$$\boldsymbol{Z} := \frac{1}{\boldsymbol{\gamma}} \log\left(\frac{\boldsymbol{\gamma}}{\boldsymbol{\sigma}}\boldsymbol{X} + 1\right), \tag{3.7}$$

where the random vector $\boldsymbol{X}$ has the density in (3.6), and operations are elementwise. As a result, the standardised density can be expressed as

$$h_{\boldsymbol{U}}(\boldsymbol{z}) = \mathbb{1}\{\max(\boldsymbol{z}) > 0\}\frac{1}{\mathbb{E}[\exp(\max(\boldsymbol{U}))]}\int_0^\infty f_{\boldsymbol{U}}(\boldsymbol{z} + \log t)\mathrm{d}t. \tag{3.8}$$

In practice, the mGPD is usually first defined on the standardised scale given in (3.8) and then transformed back to the observation scale using (3.7). Note that the U density in (3.6) or (3.8) still lacks finite parametrisation since it depends on the density function $f_{\boldsymbol{U}}$, which determines the extremal dependence of the mGPD. Here, we assume $f_{\boldsymbol{U}}$ to have independent reverse exponential components, that is $f_{\boldsymbol{U}}(\boldsymbol{x}) = \prod_{j=1}^d a_j^{-1}\exp({a_j}^{-1}x_j)$, for $x_j \in (-\infty, 0), a_j > 0$. Other forms of the $f_{\boldsymbol{U}}$ and generators are briefly discussed in Section 3.6. In the reverse exponential case, (3.8) can be explicitly expressed as

$$h_{\boldsymbol{U}}(\boldsymbol{z}) = \frac{\exp[-\max(\boldsymbol{z})(1 + \sum_{j=1}^d a_j^{-1})]}{\mathbb{E}[\exp(\max(\boldsymbol{U}))]}\frac{\prod_{i=1}^d a_j^{-1}\exp(a_j^{-1}z_j)}{1 + \sum_{i=1}^d a_j^{-1}}, \tag{3.9}$$

where $\mathbb{E}[\exp(\max(\boldsymbol{U}))] = \int_0^\infty 1 - \mathbb{P}\left(\exp(\max(\boldsymbol{U})) \le t\right)\mathrm{d}t = (\sum_{j=1}^d a_j^{-1})/(1 + \sum_{i=1}^d a_j^{-1})$.

3.2.2 Multivariate bulk distribution

The mGPD in Section 3.2.1 is supported on the complement of the negative orthant. Introducing the threshold vector $\boldsymbol{u}$ shifts this support to the region $A = \{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{x} - \boldsymbol{u} \nleq \boldsymbol{0}\}$, that is, the set of points where at least one component exceeds its corresponding threshold. On the complement $\mathbb{R}^d \setminus A$, we model observations with a multivariate bulk distribution F_{bulk} on $\mathbb{R}^d$, having density f_{bulk}. The choice of F_{bulk} is flexible and should be guided by the characteristics of the system under study. The only requirement is that both f_{bulk} and F_{bulk} can be evaluated exactly over their support. This condition is met, for example, by a multivariate normal distribution or by copula-based constructions with specified marginals (see Section 3.6). For illustration, in the remainder of this paper, we take F_{bulk} to be multivariate normal, with density

$$f_{\text{bulk}}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = (2\pi)^{-d/2}(\det\boldsymbol{\Sigma})^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}, \quad \boldsymbol{x}\leq\boldsymbol{u}, \tag{3.10}$$

where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix.

3.2.3 Multivariate extreme mixture distribution

Since the tail and bulk define a partition of the mixture model's support, combining (3.10) and (3.9), the density function of our multivariate extreme mixture model can be written as

$$f(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{a},\boldsymbol{\sigma},\boldsymbol{\gamma},\boldsymbol{u}) = \begin{cases} f_{\text{bulk}}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}), & \text{if } \boldsymbol{x}\leqslant\boldsymbol{u} \\ \\ [1-F_{\text{bulk}}(\boldsymbol{u}|\boldsymbol{\mu},\boldsymbol{\Sigma})]h_{\boldsymbol{U}}(\boldsymbol{x}-\boldsymbol{u}|\boldsymbol{a},\boldsymbol{\sigma},\boldsymbol{\gamma}), & \text{otherwise.} \end{cases} \tag{3.11}$$

The density in (3.11) indicates that data are characterised by the multivariate normal distribution when all components are less than the threshold $\boldsymbol{u}$ and described by the U-representation of the mGPD with a reverse exponential generator when at least one component exceeds the threshold. Equation (3.11) can also be expressed in the standard mixture form by introducing the truncated bulk density $f^*_{\text{bulk}}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathbb{1}\{\boldsymbol{x}\leq\boldsymbol{u}\}f_{\text{bulk}}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})/F_{\text{bulk}}(\boldsymbol{u}|\boldsymbol{\mu},\boldsymbol{\Sigma})$. In this notation, our extreme mixture model becomes

$$f(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{a},\boldsymbol{\sigma},\boldsymbol{\gamma},\boldsymbol{u}) = \pi f^*_{\text{bulk}}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) + (1-\pi)h_{\boldsymbol{U}}(\boldsymbol{x}-\boldsymbol{u}|\boldsymbol{a},\boldsymbol{\sigma},\boldsymbol{\gamma}),$$

where the mixture probability is $\pi = F_{\text{bulk}}(\boldsymbol{u}|\boldsymbol{\mu},\boldsymbol{\Sigma})$. This form is particularly convenient for sampling from (3.11) when f^*_{bulk} can be sampled directly.

To illustrate our model, Figure 3.1 shows a two-dimensional representation of the density in (3.11). The dashed lines at $X_1 = u_1$ and $X_2 = u_2$ mark the boundary between the bulk and tail regions, as well as the discontinuities in the density. These discontinuities are visible as abrupt changes in the contour lines when crossing the threshold. Although the sharp change in density at the threshold may appear artificial, its practical effect is minimal for small sample sizes. Figure 3.1 also helps to interpret the marginal densities. For example, in the left-hand panel, the marginal density $f_1(x_1)$ behaves as follows. For $X_1 > u_1 = 5.7$, the tail region is modelled entirely by the bivariate GPD, so the conditional marginal $f_1(x_1 \mid x_1 > u_1)$ reduces

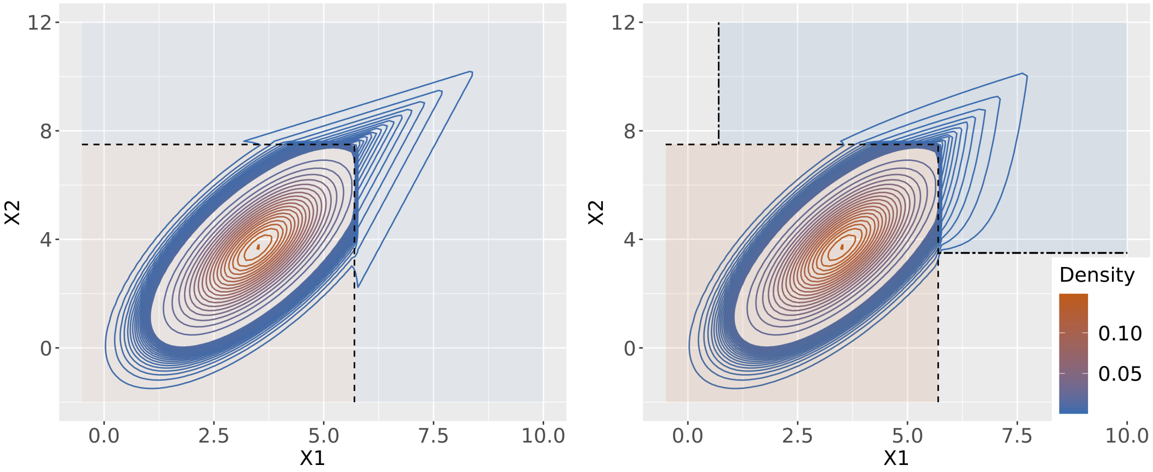

Figure 3.1: Two-dimensional density contour plots derived from (3.11). Dashed lines indicate threshold values, whereas the double-dashed lines in the right plot highlight the lower endpoints of the bivariate GPD. Shaded areas represent the support of each mixture component. For the left plot, the bivariate GPD parameters are specified as $\boldsymbol{\sigma} = \mathbf{1}$, $\boldsymbol{\gamma} = \mathbf{0}$, and $\boldsymbol{a} = (1, 2)$. Conversely, the right plot employs parameters $\boldsymbol{\sigma} = (1, 1.2)$, $\boldsymbol{\gamma} = (0.2, 0.3)$, and $\boldsymbol{a} = (1, 2)$. Both plots share common parameters for the bulk and threshold, represented by $\boldsymbol{\mu} = (3.5, 3.7)$, $\boldsymbol{\Sigma} = \left(\begin{smallmatrix} 1 & 1.05 \\ 1.05 & 2.25 \end{smallmatrix}\right)$, and $\boldsymbol{u} = (5.7, 7.5)$.

to a univariate GPD. For $X_1 < u_1$, the marginal density $f_1(x_1 \mid x_1 < u_1)$ is obtained by integrating x_2 out of a mixture of the bivariate GPD (triangular contours) and the bivariate normal distribution (elliptical contours). In this sense, our framework extends the univariate approach of do Nascimento et al. (2012) to the multivariate case, with both methods combining a mixture of parametric bulk distributions with a univariate GPD for the tail in the marginal analysis.

As shown in (3.11), the tail behaviour of our model is entirely determined by the mGPD. Consequently, our multivariate extreme mixture model inherits all theoretical properties of the mGPD, including threshold stability, GPD conditional margins, and sum-stability under shape constraints (Kiriliouk et al., 2019). In particular, our model lies in the max-domain of attraction of an mGEVD, as stated in Proposition 8, a result that follows directly from Theorem 2.2 of Rootzén and Tajvidi (2006) and the fact that the tail of our model is an mGPD. We include the proof in Section 3.8.2 of the Supplementary Materials for completeness.

As a consequence of being in the max-domain of attraction of an mGEVD, our model is always asymptotically dependent; that is, the coefficient

$$\chi := \lim_{r \to 1^-} \chi(r), \qquad \chi(r) = \frac{\mathbb{P}\{\bigcap_{j=1}^{d} \{X_j > F_j^{-1}(r)\}\}}{1 - r}, \tag{3.12}$$

is strictly positive. This implies that the model may introduce bias when the underlying data exhibit asymptotic independence (i.e., $\chi = 0$ as $r \to 1^-$). In practice, however, the distinction between weak asymptotic dependence and complete asymptotic independence is often ambiguous, particularly in low-dimensional and small-sample settings. Our simulation results demonstrate that the model can still provide reasonable estimates even when the data are asymptotically independent.

Proposition 8. *The distribution F with density* (3.11) *lies in the max-domain of attraction of an mGEVD, regardless of the choice of bulk distribution.*

3.3 Bayesian inference

Although maximum likelihood inference could, in principle, be performed using methods such as the EM algorithm for mixture models, the Bayesian framework provides a natural means of quantifying all sources of uncertainty in our model, particularly for the threshold vector. Here, we detail the prior specification for all model parameters: $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for the bulk; the threshold vector $\boldsymbol{u}$; and $\boldsymbol{a}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\sigma}$ for the tail.

3.3.1 Priors for the parameters in the bulk

A standard approach for assigning priors in the multivariate normal setting is to use a multivariate normal prior for the mean vector and an inverse-Wishart prior for the covariance matrix. While this choice offers conjugacy, it can be overly restrictive for the covariance matrix, as noted by Sun and Berger (2007). To address this limitation, Barnard et al. (2000) proposed decomposing the covariance matrix as $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{S})\boldsymbol{C}\text{diag}(\boldsymbol{S})$, where $\boldsymbol{C}$ is the d-dimensional correlation matrix and $\boldsymbol{S}$ is the $d \times 1$ vector of standard deviations. Priors are then assigned separately to these components. In our simulations and case study (Sections 3.4 and 3.5), we assign independent Half-Normal(h_{s_j}) priors to each s_j in $\boldsymbol{S}$, choosing h_{s_j} sufficiently large to be noninformative; specifically, h_{s_j} is set to 50 times the scale of the data. For the correlation matrix, we adopt the Lewandowski–Kurowicka–Joe (LKJ) prior (Lewandowski et al., 2009), given by

$$\pi(\boldsymbol{C}) \propto (\det \boldsymbol{C})^{\delta-1},$$

where the parameter δ controls the strength of correlation. When $\delta = 1$, the prior is uniform over all valid d-dimensional correlation matrices. $\delta > 1$ favors stronger correlations (larger diagonal elements), while $\delta < 1$ favors weaker correlations. For computational considerations, the LKJ prior can also be defined on the upper-triangular Cholesky factor $\boldsymbol{L}$ of $\boldsymbol{C}$ using a change of variable. For the mean vector $\boldsymbol{\mu}$, we adopt independent normal priors $\mu_j \sim N(m_j, t_j^2)$ for $j = 1, \ldots, d$. In summary, the joint prior for the bulk parameters is

$$\pi(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \pi(\boldsymbol{\mu}, \boldsymbol{S}, \boldsymbol{L}) \propto \left|\boldsymbol{L}^\top \boldsymbol{L}\right|^{\delta} \prod_{j=1}^{d} \varphi\left(\frac{\mu_j - m_j}{t_j}\right) \exp\left\{-\frac{s_j^2}{2h_{s_j}^2}\right\} \mathbb{1}(s_j > 0) \tag{3.13}$$

where φ denotes the standard normal density.

3.3.2 Prior for the threshold

The choice of threshold plays a critical role in the threshold exceedance framework. A high threshold results in too few exceedances for reliable inference, leading to increased variance in parameter estimates. Conversely, a low threshold may violate the asymptotic assumptions of the mGPD, thereby introducing bias. We seek to retain thresholds in the upper quantiles while allowing sufficient flexibility for the data to determine their optimal values and to quantify the associated uncertainty. To this end, we parametrise the threshold u_j using a marginal reference location at the τ-quantile, denoted by $q_{\tau,j}$, together with a positive offset. Specifically,

$$u_j = q_{\tau,j} + o_j, \quad o_j \overset{\text{i.i.d.}}{\sim} \text{Half-Normal}(h_o), \tag{3.14}$$

where h_o is the scale parameter of the half-normal distribution, controlling how far the threshold may deviate from the reference level. Although $q_{\tau,j}$ is data-dependent, our substantive prior is placed on the uncertainty of exceedances and is itself data-independent.

The prior in (3.14) can be viewed as a location-aware extension of the truncated normal prior for thresholds (Behrens et al., 2004; MacDonald et al., 2011; do Nascimento et al., 2012), enabling its application to datasets that may include negative values. An alternative approach is to impose a discrete prior on upper order statistics (de Zea Bermudez et al., 2001; Behrens et al., 2004). However, in our framework, this choice is more prone to submixing issues during Markov chain Monte Carlo (MCMC) inference than in univariate settings (Behrens et al., 2004; MacDonald et al., 2011; do Nascimento et al., 2012). We therefore adopt a continuous prior on the threshold to preserve flexibility and facilitate efficient MCMC sampling.

Assuming independence across marginal priors, the joint prior distribution of the threshold vector $\boldsymbol{u}$ is given by

$$\pi(\boldsymbol{u}) \propto \prod_{j=1}^{d} \exp\left\{-\frac{(u_j - q_{\tau,j})^2}{2h_o^2}\right\} \mathbb{1}(u_j > q_{\tau,j})$$

3.3.3 Priors for the parameters in the tail

The marginal parameters of the mGPD are $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_d)$ and $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_d)$, whose interpretations mirror those in the univariate GPD, since each marginal conditional distribution $H_j(x_j \mid x_j > 0)$ is univariate GPD. Various priors for GPD parameters have been proposed, including quantile-based priors (Coles and Powell, 1996), Jeffreys priors (Castellanos and Cabras, 2007), and penalised complexity priors (Opitz et al., 2018b). In our framework, we impose weakly informative priors with hard constraints to preserve essential tail properties, for example, ensuring finite marginal excess expectations by restricting $\gamma_j < 1$, while avoiding further parameter preference. Specifically, we assume $\gamma_j \overset{\text{i.i.d.}}{\sim}$ Uniform(l_γ, r_γ) and $\sigma_j \overset{\text{i.i.d.}}{\sim}$ Half-Norm(h_σ), where $l_{(\cdot)}$ and $r_{(\cdot)}$ denote lower and upper bounds, respectively. The components of the tail dependence parameter $\boldsymbol{a} = (a_1, \ldots, a_d)$, defined in (3.9), are independently assigned prior Half-Norm(h_a). The joint prior for all tail parameters is therefore

$$\pi(\boldsymbol{a}, \boldsymbol{\gamma}, \boldsymbol{\sigma}) \propto \prod_{j=1}^{d} \exp\left\{-\frac{a_j^2}{2h_a^2} - \frac{\sigma_j^2}{2h_\sigma^2}\right\} \mathbb{1}(a_j > 0) \cdot \mathbb{1}(\sigma_j > 0) \cdot \mathbb{1}(l_\gamma < \gamma_j < r_\gamma). \tag{3.15}$$

3.3.4 Posterior inference

Let $\boldsymbol{\theta}_b = (\boldsymbol{\mu}, \boldsymbol{S}, \boldsymbol{L})$, and $\boldsymbol{\theta}_t = (\boldsymbol{a}, \boldsymbol{\gamma}, \boldsymbol{\sigma})$. Let $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)^T$ be an $n \times d$ sample matrix. Define the mutually exclusive sets B and T to correspond to the rows of $\boldsymbol{X}$ classified as bulk and tail data, respectively, with $B, T \subset \{1, \ldots, n\}$. The log posterior density of the multivariate extreme mixture model in (3.11) is given by:

$$
\begin{aligned}
\log \pi(\boldsymbol{\theta}_b, \boldsymbol{u}, \boldsymbol{\theta}_t | \boldsymbol{X}) \propto \sum_{i=1}^{n} & \log \{ f_{\text{bulk}}(\boldsymbol{x_i} | \boldsymbol{\theta}_b) \cdot \mathbb{1}(\boldsymbol{x_i} \leq \boldsymbol{u}) \\
& + [1 - F_{\text{bulk}}(\boldsymbol{u} | \boldsymbol{\theta}_b)] f_{\text{tail}}(\boldsymbol{x_i} - \boldsymbol{u} | \boldsymbol{\theta}_t) \cdot \mathbb{1}(\boldsymbol{x_i} \nleq \boldsymbol{u}) \} \\
& + \log \pi(\boldsymbol{\theta}_b) + \log \pi(\boldsymbol{u}) + \log \pi(\boldsymbol{\theta}_t) \\
\propto \sum_{i \in B} & \log f_{\text{bulk}}(\boldsymbol{x_i} | \boldsymbol{\theta}_b) + \sum_{i \in T} \log f_{\text{tail}}(\boldsymbol{x_i} - \boldsymbol{u} | \boldsymbol{\theta}_t) + |T| \log[1 - F_{\text{bulk}}(\boldsymbol{u} | \boldsymbol{\theta}_b)] \\
& + \log \pi(\boldsymbol{\theta}_b) + \log \pi(\boldsymbol{u}) + \log \pi(\boldsymbol{\theta}_t),
\end{aligned}
\tag{3.16}
$$

where $|T| := |T(\boldsymbol{u})|$ denotes the number of tail observations, which depends on the threshold.

We perform inference using Markov chain Monte Carlo (MCMC) sampling from (3.16). As an initial step, we employ a multivariate random-walk Metropolis–Hastings (MH) algorithm with independent normal proposals for each parameter. Proposal scaling is adaptively tuned to achieve the theoretical optimal acceptance rate (Roberts and Rosenthal, 2001). However, this approach can result in suboptimal mixing for the threshold parameters $\boldsymbol{u}$ and certain components of $\boldsymbol{\theta}_t$, due to strong posterior dependence among parameters. Specifically, the threshold stability property of the mGPD (Kiriliouk et al., 2019) implies that increasing the threshold of an mGPD with shape parameter $\boldsymbol{\gamma}$ and scale parameter $\boldsymbol{\sigma}$ by $\boldsymbol{w}$ yields a new mGPD with identical marginal shape and tail dependence structure, but with an updated scale parameter $\boldsymbol{\sigma} + \boldsymbol{\gamma w}$. This induces strong dependence between $\boldsymbol{u}$ and $\boldsymbol{\sigma}$ in the posterior distribution.

To address this issue, we define $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma} - \boldsymbol{\gamma u}$, which is invariant for sufficiently high $\boldsymbol{u}$, and reparametrise $(\boldsymbol{\sigma}, \boldsymbol{\gamma}, \boldsymbol{u})$ as $(\tilde{\boldsymbol{\sigma}}, \boldsymbol{\gamma}, \boldsymbol{u})$. Retaining the original priors for $\boldsymbol{\sigma}$, $\boldsymbol{\gamma}$, and $\boldsymbol{u}$, the induced prior for $\tilde{\boldsymbol{\sigma}}$ is

$$
\pi(\tilde{\boldsymbol{\sigma}} \mid \boldsymbol{u}, \boldsymbol{\gamma}) = \pi_{\boldsymbol{\sigma}}(\tilde{\boldsymbol{\sigma}} + \boldsymbol{\gamma u}) \cdot |1| \propto \prod_{j=1}^{d} \exp\left\{ -\frac{(\tilde{\sigma}_j + \gamma_j u_j)^2}{2h_\sigma^2} \right\} \mathbb{1}(\tilde{\sigma}_j > -\gamma_j u_j)
$$

To further improve sampling efficiency, we block update the threshold $\boldsymbol{u}$ and tail parameters $(\boldsymbol{a}, \tilde{\boldsymbol{\sigma}}, \boldsymbol{\gamma})$ using the automated factor slice sampler (AFSS; Tibbits et al., 2014). The AFSS mitigates posterior dependence by performing univariate slice sampling along the eigenvectors of the covariance matrix obtained from a quadratic approximation of the target distribution. Empirically, this approach substantially improves computational efficiency, often by more than an order of magnitude when measured by the ratio of effective sample size to runtime. The remaining bulk parameters are updated using MH with a Gaussian random-walk proposal. All AFSS-based MCMC routines are implemented using the `NIMBLE` package in R (de Valpine et al., 2017). For consistency with the parametrisation in (3.11), we report results using the original parametrisation rather than the reparametrised form.

3.4 Simulations

We conduct simulation studies to evaluate: (i) the ability of our model to recover bulk, tail, and threshold parameters under diverse tail behaviours; (ii) robustness to model misspecification, particularly when the true process is asymptotically independent; and (iii) its capacity to capture key dependence measures χ, $\bar{\chi}$ and Kendall's τ. These aspects are essential both for accurate extrapolation to unseen observations and for a proper characterisation of the bulk data. For computational tractability, the dimensionality is restricted to two.

3.4.1 Well-specified scenarios (Scenario 1)

In the first experiment, we assess parameter recovery under correctly specified models with three different marginal tail types, detailed in Table 3.1. The bulk parameters and thresholds are fixed across scenarios, as shown in the "True Value" (TV) columns of Table 3.2. Each dataset contains $n = 2000$ observations, with approximately 5% classified as tail data.

Table 3.1: Tail shape parameters for the three well-specified simulation scenarios. Each scenario differs in the heaviness of the marginal tails, as determined by the shape parameters (γ_1, γ_2).

Scenario	γ_1	γ_2	Description
1.1	0.3	0.1	Both margins heavy-tailed
1.2	0.2	-0.2	One heavy-, one light-tailed
1.3	-0.1	-0.3	Both margins light-tailed

We adopt weakly informative priors. For the bulk prior in (3.13), we set $\delta = 1.3$, $m_1 = m_2 = 0$, and $h_{s_1} = h_{s_2} = 50$. The threshold vector is constrained to lie above the 0.8-quantile of each margin, with scale parameter $h_o = 10$. For the tail parameters in (3.15), we specify $h_a = h_\sigma = 50$ to reflect limited prior information on the scale in both the margins and $f_{\boldsymbol{U}}$. The shape parameter bounds are set to $l_\gamma = -1$ and $r_\gamma = 1$ to ensure a finite mean for the GPD of the marginal threshold excesses.

Table 3.2: Average posterior means with 95% credible interval (CI) lengths in brackets, and coverage rates (CR) for all parameters across three well-specified scenarios. Each scenario uses $n = 2000$ observations with approximately 5% classified as tail data. The "True Value" (TV) column gives the parameter values used for data generation. Results are averaged over 1000 simulation replicates. CI length is the average length of the marginal 95% credible interval across replicates. CR is the proportion of replicates in which the true value falls within the CI. Bulk, tail, and threshold parameters are grouped separately for clarity.

	Scenario 1.1			**Scenario 1.2**			**Scenario 1.3**		
	TV	EST	CR	TV	EST	CR	TV	EST	CR
Tail parameters									
a_1	0.5	0.55 (0.52)	0.95	0.5	0.54 (0.47)	0.95	0.5	0.53 (0.45)	0.94
a_2	1.2	1.23 (0.86)	0.95	1.2	1.25 (0.83)	0.96	1.2	1.25 (0.82)	0.96
σ_1	0.5	0.52 (0.30)	0.95	0.5	0.51 (0.29)	0.94	0.5	0.51 (0.28)	0.95
σ_2	1.2	1.25 (0.66)	0.95	1.2	1.23 (0.64)	0.96	1.2	1.23 (0.64)	0.95
γ_1	0.3	0.31 (0.40)	0.96	0.2	0.22 (0.39)	0.95	-0.1	-0.09 (0.35)	0.95
γ_2	0.1	0.09 (0.25)	0.95	-0.2	-0.20 (0.30)	0.96	-0.3	-0.30 (0.32)	0.95
Threshold parameters									
u_1	5.5	5.51 (0.21)	0.95	5.5	5.50 (0.14)	0.94	5.5	5.50 (0.12)	0.95
u_2	6.7	6.70 (0.08)	0.94	6.7	6.70 (0.08)	0.95	6.7	6.70 (0.09)	0.95
Bulk parameters									
$L[1,2]$	0.70	0.70 (0.05)	0.96	0.70	0.70 (0.05)	0.96	0.70	0.70 (0.05)	0.96
$L[2,2]$	0.71	0.71 (0.05)	0.96	0.71	0.71 (0.05)	0.96	0.71	0.71 (0.05)	0.96
μ_1	3.5	3.50 (0.09)	0.97	3.5	3.50 (0.09)	0.98	3.5	3.50 (0.09)	0.98
μ_2	4.0	4.00 (0.13)	0.98	4.0	4.00 (0.13)	0.98	4.0	4.00 (0.13)	0.98
s_1	1.0	1.00 (0.07)	0.97	1.0	1.00 (0.07)	0.96	1.0	1.00 (0.07)	0.96
s_2	1.5	1.50 (0.10)	0.97	1.5	1.50 (0.10)	0.98	1.5	1.50 (0.10)	0.98

Posterior samples are obtained using the AFSS sampler with three parallel chains, each of length 30,000, a burn-in of 20,000, and thinning by a factor of 10. Each scenario is replicated 1,000 times to assess estimation variability. Table 3.2 reports average posterior means, the average lengths of 95% credible intervals (in brackets), and coverage rates (CR). Across all parameters and scenarios, CR values are close to 0.95, indicating accurate recovery of both marginal and dependence parameters.

3.4.2 Misspecification scenario (Scenario 2)

We now assess robustness under departures from the model's asymptotic dependence assumption. This is relevant because our model is always asymptotically dependent ($\chi > 0$), and applying it to asymptotically independent data can, in principle, induce bias.

We generate data from a bivariate normal distribution, which is asymptotically independent ($\chi = 0$; Sibuya 1960). Bulk parameters match those in Table 3.2. We compare our bivariate extreme mixture model (BEMM) to the bivariate mixture copula model (BMCM) of André et al., using their best-performing configuration for Gaussian data: a Student-t copula for the bulk and an inverted Gumbel copula for the tail. For each method, we assess overall dependence through the Kendall's rank coefficient, defined as

$$\tau = 2\mathbb{P}((X_1 - X_1')(X_2 - X_2') > 0)$$

for a random pair (X_1, X_2) and its independent replicate (X_1', X_2'), with marginal distribution F_1 and F_2. For tail dependence, we use two measures. The first is χ, defined in (3.12), which distinguishes between asymptotic dependence ($\chi > 0$) and asymptotic independence ($\chi = 0$). For our model, based on the mGPD in (3.9), χ has the closed-form expression (see Section 3.8.2 of the Supplementary Materials for derivation):

$$\chi = 1 - \left(\frac{1 + a_{(1)}}{1 + a_{(2)}}\right)^{1 + a_{(2)}^{-1}} \frac{a_{(2)}}{a_{(1)}} \frac{a_1 a_2}{a_1 a_2 + a_1 + a_2}, \tag{3.17}$$

where $a_{(1)} = \min\{a_1, a_2\}$, $a_{(2)} = \max\{a_1, a_2\}$. This expression further shows that our model is asymptotically dependent, as $\chi > 0$ for all $a_1, a_2 > 0$.

While $\chi(r)$ is effective for distinguishing between asymptotic dependence and independence, it provides limited information on the strength of asymptotic independence. To address this, we also use $\bar{\chi}$ (Coles et al., 1999), defined as

$$\bar{\chi} = \lim_{r \to 1^-} \bar{\chi}(r), \qquad \bar{\chi}(r) = \frac{2 \log \mathbb{P}(F_1(X_1) > r)}{\log \mathbb{P}(F_1(X_1) > r), F_2(X_2) > r)} - 1,$$

which differentiates among asymptotically independent cases. $\bar{\chi}$ takes values in $[-1, 1)$, increases with the strength of extremal dependence, and approaches 1 for asymptotically dependent distributions.

For comparability with BMCM's frequentist inference, we obtain point estimates for the BEMM by averaging the empirical dependence metrics (χ, $\bar{\chi}$, and Kendall's τ) computed from 3,000 replicated datasets, each containing 2,000 observations. These replicated datasets, denoted $\boldsymbol{y}_{\text{rep}}$, are drawn from the posterior predictive distribution:

$$p(\boldsymbol{y}_{\text{rep}} | \boldsymbol{X}) = \int p(\boldsymbol{y} | \boldsymbol{\theta}) p(\boldsymbol{\theta} | \boldsymbol{X}) \mathrm{d}\boldsymbol{\theta}, \tag{3.18}$$

where $p(\boldsymbol{y}|\boldsymbol{\theta})$ represents the density function in (3.11), and $p(\boldsymbol{\theta}|\boldsymbol{X})$ is the posterior distribution.

Figure 3.2 presents boxplots of $\chi(r)$, $\bar{\chi}(r)$, and Kendall's τ from 100 BMCM experiments and 1,000 BEMM experiments. The number of BMCM experiments matches those in André et al., while the number of BEMM experiments is increased to account for the additional parameter uncertainty arising from prior specification. For tail dependence, both models perform similarly for $r \leq 0.95$. At $r = 0.99$, the BEMM slightly overestimates, and the BMCM slightly underestimates these quantities; in both cases, true values (black dots) lie within the interquartile range. Kendall's τ estimates are comparable across models, reflecting its lower sensitivity to tail misspecification.

Overall, both simulation studies show that the BEMM delivers accurate parameter estimation under correct specification, maintains nominal credible-interval coverage, and yields reasonable dependence estimates even under asymptotic independence. These results suggest that the model's asymptotic dependence property is not unduly problematic in moderate-sample, low-dimensional settings, where the empirical distinction between weak dependence and independence is inherently blurred.

3.5 Application

The frequency of heatwaves in the United Kingdom has increased markedly in recent years, with the 2022 event setting a record temperature of 40.3°C and causing substantial societal impacts,

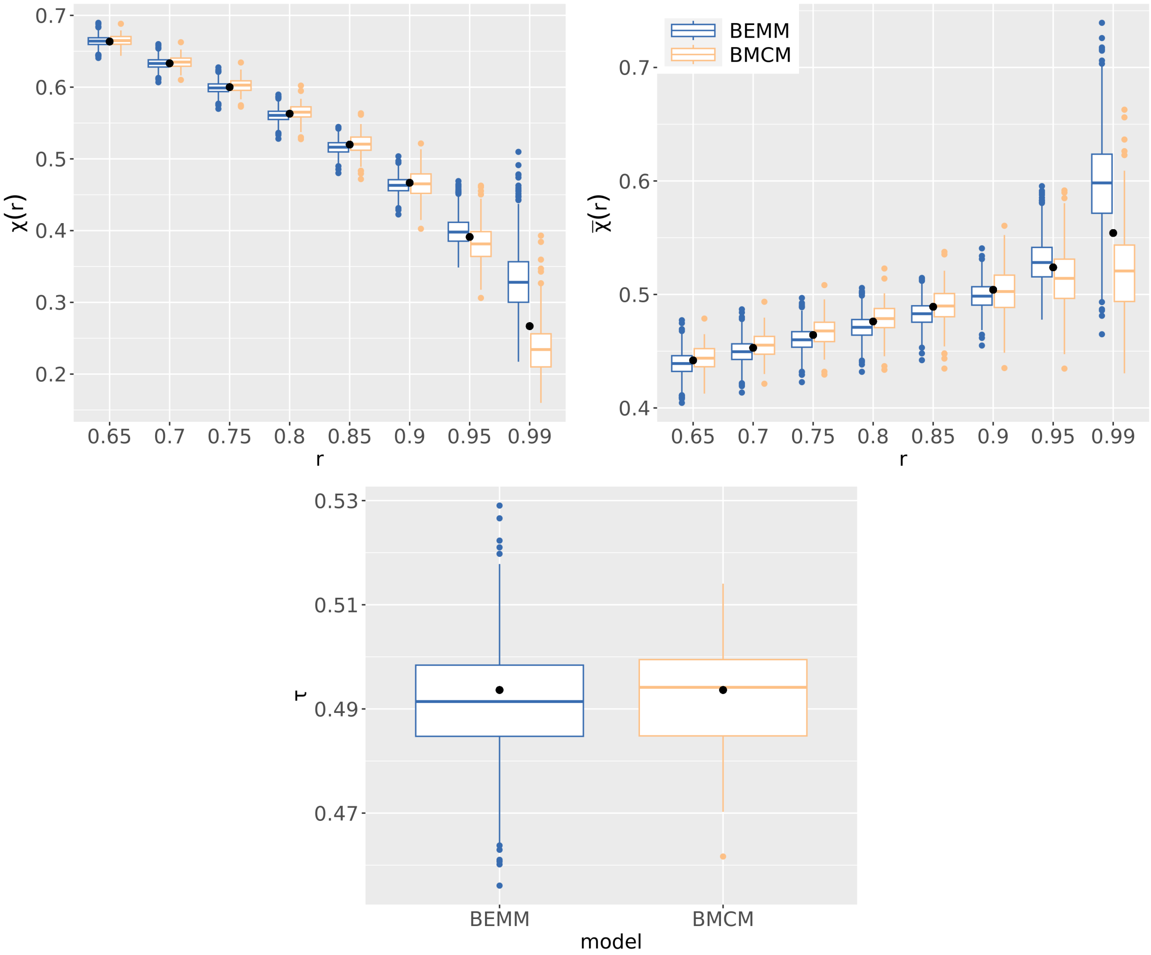

Figure 3.2: Boxplots comparing extremal dependence metrics ($\chi(r)$, $\bar{\chi}(r)$), and Kendall's τ for the bivariate extreme mixture model (BEMM, blue) and the bivariate mixture copula model (BMCM, orange) under Scenario 2 (Gaussian data, asymptotic independence). Yellow boxes are based on estimates from 100 BMCM experiments. Blue boxes correspond to empirical metrics from 1,000 BEMM experiments, where each experiment reports the point estimates of the three metrics as averages over 3,000 replicated datasets. Black points denote the true values of the dependence metrics.

including an estimated 3,000 excess deaths. To study temperature behaviour during such extremes and compare patterns across regions, we analyse daily maximum temperatures from two stations: Bishop's Lane (Ringmer, East Sussex) and Model Farm (Shirburn, Oxfordshire). Our aim is to quantify joint tail risk while accounting for uncertainty in both thresholds and dependence parameters. The data, obtained from the Centre for Environmental Data Analysis[1], cover 2016-2021, yielding 1,945 daily maximum temperature observations after removing records with missing values.

3.5.1 Pre-processing and model specification

Daily temperatures exhibit strong seasonality and short-term dependence. We account for these by including sinusoidal functions of time to model the annual cycle and a first-order autoregressive term to capture temporal correlation. Specifically, for the air temperature $Y_{t,j}$ on day t at site $j \in 1, 2$, we assume

$$Y_{t,j} = \beta_{0,j} + \beta_{1,j} \sin\left(\frac{2\pi}{365}t\right) + \beta_{2,j} \cos\left(\frac{2\pi}{365}t\right) + \beta_{3,j} Y_{t-1,j} + \varepsilon_j, \tag{3.19}$$

where $\beta_{i,j}$ $(i = 0, 1, 2, 3)$ are regression coefficients and ε_j is a noise term. Stationarity of the residuals is verified using autocorrelation plots (provided in Section 3.8.3 of the Supporting Materials). The joint negative residual $\boldsymbol{E} = -(\varepsilon_1, \varepsilon_2)$, as shown in Figure 3.3, presents heavy right tails and strong extremal dependence. To characterise these joint residuals, we fit our BEMM model to $\boldsymbol{E}$ using the priors from the simulation study, with the additional constraint $\gamma_j + 1/a_j \geq 0$ $(i = 1, 2)$ to ensure finite marginal expectations. Finite marginal expectations enable the use of proper scoring rules for model evaluation, as discussed at the end of this section. The derivation of these constraints is provided in Section 3.8.2 of the Supplementary Materials. We run three parallel chains of 20,000 iterations using the hybrid MH and AFSS sampler as in the simulations, and discard the first 10,000 as burn-in and thinning by a factor of 10. Convergence is assessed using trace plots and the Gelman–Rubin diagnostic (reported in Section 3.8.4 of the Supplementary Materials).

[1] https://catalogue.ceda.ac.uk/uuid/dbd451271eb04662beade68da43546e1

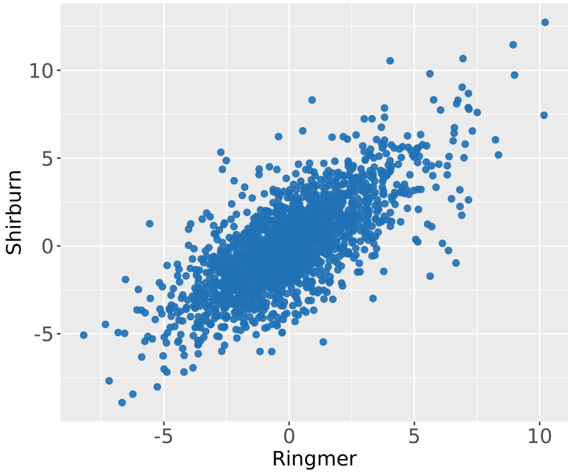

Figure 3.3: Scatterplot of the negative residuals from model (3.19).

3.5.2 Results

Thresholds. Figure 3.4 presents histograms of the posterior samples for all parameters. Notably, posterior distributions for u_1 and u_2 are multimodal. For u_1, the dominant mode occurs near the 92nd percentile, with smaller peaks at the 89th and 94th percentiles. For u_2, the histogram is left-skewed, with clusters around the 93rd, 94th, and 95th percentiles. To investigate this behaviour, we examine the joint posterior density $\pi(\boldsymbol{u} \mid \boldsymbol{X})$ together with a binned expected log-likelihood surface $A(\boldsymbol{u}) = \mathbb{E}[\log p(\boldsymbol{X} \mid \boldsymbol{\theta}_t, \boldsymbol{\theta}_b, \boldsymbol{u})]$, computed by averaging $\log p(\boldsymbol{X} \mid \boldsymbol{\theta}_t, \boldsymbol{\theta}_b, \boldsymbol{u})$ over posterior draws whose $\boldsymbol{u}$ values fall within each bin, as shown in Figure 3.5. The joint posterior exhibits three well-separated modes, and $A(\boldsymbol{u})$ reveals near-equal values across the modal regions. This indicates that the multimodality reflects genuine ambiguity in threshold selection supported by the likelihood under the UK temperature data, rather than a sampling artefact.

Marginal fit. Marginal model performance is assessed using posterior predictive checks. Specifically, we generate 3,000 replications of $\boldsymbol{E}$ from (3.18) to construct posterior predictive distributions for marginal quantiles. Figure 3.6 presents quantile–quantile (Q–Q) plots comparing posterior predictive quantiles from the replicated data to the empirical quantiles of $\boldsymbol{E}$, evaluated at uniformly spaced percentiles from 0 to 0.99. For both margins, the posterior predictive means lie close to the 45-degree line, which itself remains within the 95% credible band. For comparison, we include quantiles from a bivariate normal distribution, commonly used for residual modelling, fitted to $\boldsymbol{E}$ via maximum likelihood, though it lacks the tail correc-

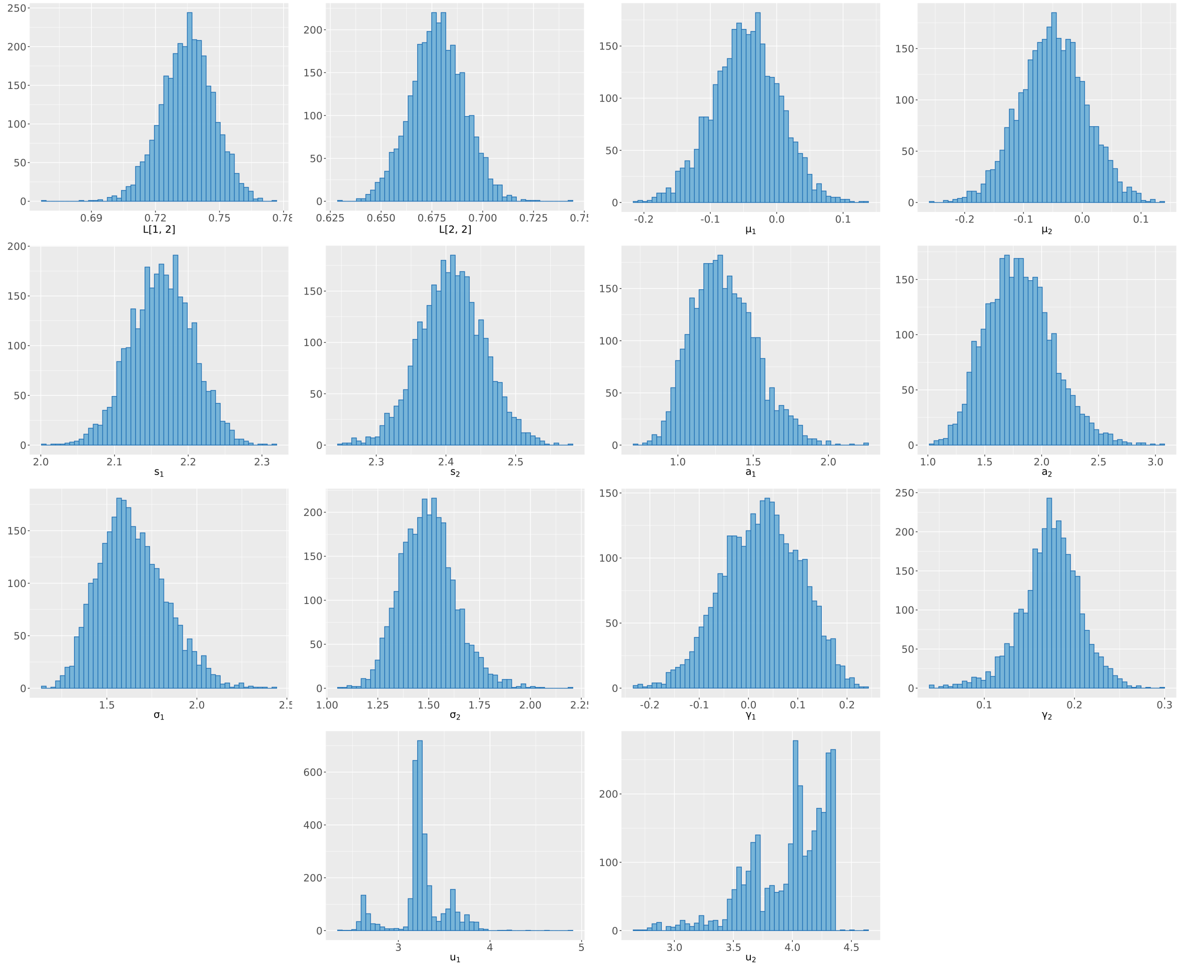

Figure 3.4: Histograms of the posterior distributions for all model parameters. The first six plots correspond to bulk parameters; the remaining plots show tail parameters and threshold components (last two plots).

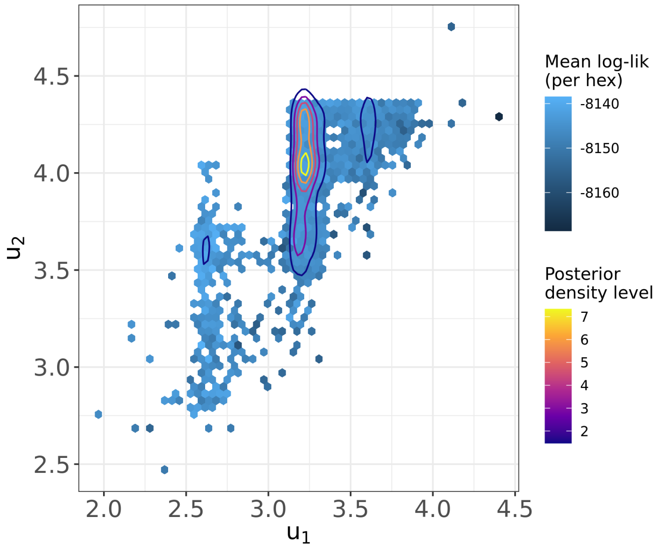

Figure 3.5: Joint posterior density of the thresholds and the binned mean log-likelihood surface. Contours represent the joint posterior density $\pi(\boldsymbol{u} \mid \boldsymbol{X})$. The background shading shows the binned mean log-likelihood, $A(\boldsymbol{u}) = \mathbb{E}[\log p(\boldsymbol{X} \mid \boldsymbol{\theta}_t, \boldsymbol{\theta}_b, \boldsymbol{u})]$, computed by averaging $\log p(\boldsymbol{X} \mid \boldsymbol{\theta}_t, \boldsymbol{\theta}_b, \boldsymbol{u})$ over posterior draws whose $\boldsymbol{u}$ values fall within each hexagonal bin.

tion offered by our mixture model. The results clearly show that our BEMM provides superior estimation, particularly for higher quantiles in both margins.

Dependence. To assess dependence, we construct posterior predictive distributions for $\chi(r)$, $\bar{\chi}(r)$, and Kendall's τ using the 3,000 replications of $\boldsymbol{E}$ generated above. Figure 3.7 shows

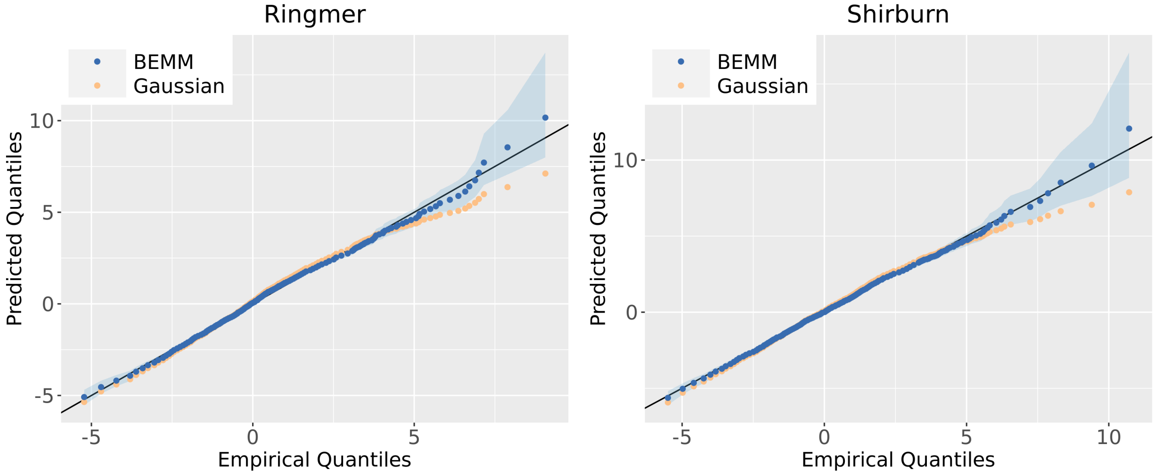

Figure 3.6: Q–Q plots of posterior predictive quantiles (blue) from 3,000 replicated datasets against the empirical quantiles. The shaded region indicates the 95% credible band. Quantiles from a fitted bivariate normal distribution (orange) using the maximum likelihood method are included for comparison.

posterior predictive distributions of $\chi(r)$, $\bar{\chi}(r)$, and Kendall's τ, alongside their empirical counterparts. The plots also include the 95% credible interval for the theoretical χ computed from (3.17). For both $\chi(r)$ and $\bar{\chi}(r)$, empirical estimates fall within the 95% credible bands for all r, except at $r = 0.68$. The theoretical χ lies near the centre of the posterior predictive $\chi(r)$ distribution, confirming the validity of (3.17). As r approaches 1, the empirical $\chi(r)$ moves toward the tail of the theoretical χ distribution, while $\bar{\chi}(r)$ approaches the posterior mean. Kendall's τ also falls within the high-density region of its posterior predictive distribution. All these suggest that the BEMM effectively captures the dependence structure.

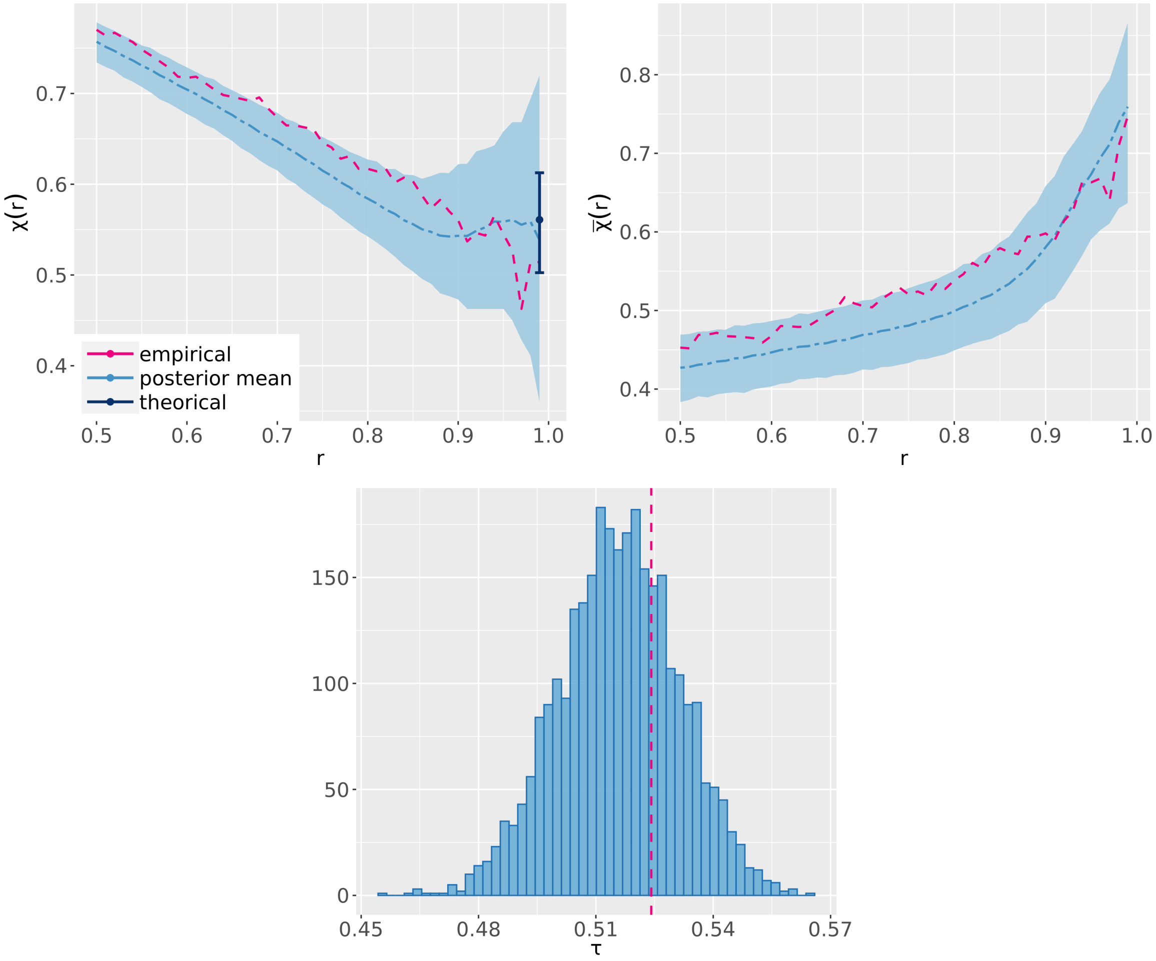

Figure 3.7: Based on 3,000 replicates from the posterior predictive distribution, the upper two plots present empirical values vs. means of the posterior prediction, each accompanied by a 95% credible band, of $\chi(r)$ and $\bar{\chi}(r)$. The error bar in the $\chi(r)$ plot represents the 95% credible interval of the theoretical χ, with the centre point indicating the posterior mean. The lower plot features a histogram of the posterior predictive Kendall's τ, with the empirical value highlighted in red.

Predictive accuracy. We further compare the predictive accuracy of the BEMM and the Gaussian model using the energy score, a multivariate generalisation of the continuous ranked probability score (CRPS; Gneiting and Raftery, 2007). For an observation $\boldsymbol{y}$ and predictive distribution F, the energy score is

$$\mathrm{ES}(F, \boldsymbol{y}) = \mathbb{E}_F \left\| \boldsymbol{X} - \boldsymbol{y} \right\| - \frac{1}{2}\mathbb{E}_F \left\| \boldsymbol{X} - \boldsymbol{X}' \right\|, \tag{3.20}$$

where $\boldsymbol{X}$ and $\boldsymbol{X}'$ are independent draws from F, and $\|\cdot\|$ denotes the Euclidean norm. The score measures the expected distance between predictions and observations, adjusted for the spread of the predictive distribution, and is proper when F has a finite expectation.

When specific regions of the distribution are of primary interest, such as the tail, threshold-weighted CRPS is often used. This approach extends to the multivariate setting through the threshold-weighted energy score (twES; Allen et al., 2023):

$$\mathrm{twES}(F, \boldsymbol{y}; v) = \mathbb{E}(\|v(\boldsymbol{X}) - v(\boldsymbol{y})\|) - \frac{1}{2}\|v(\boldsymbol{X}) - v(\boldsymbol{X}')\|$$

where $v : \mathbb{R}^d \to \mathbb{R}^d$ is a chaining function derived from a weight function $w : \mathbb{R}^d \to \mathbb{R}$ specifying the region of interest. We consider two weighting schemes:

1. **High-tail indicator weight**: $w_1(\boldsymbol{z}) = \mathbb{1}\{z_1 > q_{1,0.9}, z_2 > q_{2,0.9}\}$, where $q_{j,0.9}$ is the 0.9-quantile of margin j of $\boldsymbol{E}$. The corresponding chaining function is

$$v_1(\boldsymbol{z}) = \begin{cases} \boldsymbol{z} & \text{if} \quad w_1(\boldsymbol{z}) = 1 \\ \boldsymbol{x}_0 & \text{if} \quad w_1(\boldsymbol{z}) = 0 \end{cases}$$

for some fixed $\boldsymbol{x}_0 \in \mathbb{R}^2$.

2. **Gaussian CDF weight**: $w_2(\boldsymbol{z})$ is defined as the CDF of a bivariate normal distribution fitted to $\boldsymbol{E}$. With estimated means $\hat{\mu}_1$, $\hat{\mu}_2$ and standard deviations $\hat{s}_1$, $\hat{s}_2$, the chaining function is

$$v_2(\boldsymbol{z}) = \left((z_1 - \hat{\mu}_1)\Phi\left(\frac{z_1 - \hat{\mu}_1}{\hat{s}_1}\right) + \hat{s}_1\phi\left(\frac{\hat{z}_1 - \hat{\mu}_1}{\hat{s}_1}\right), (\hat{z}_2 - \hat{\mu}_2)\Phi\left(\frac{z_2 - \hat{\mu}_2}{\hat{s}_2}\right) + \hat{s}_2\phi\left(\frac{z_2 - \hat{\mu}_2}{\hat{s}_2}\right)\right),$$

where Φ and ϕ are the standard normal CDF and PDF, respectively.

Table 3.3 reports the unweighted energy scores and their threshold-weighted variants for both the BEMM and Gaussian models. Across all metrics, the BEMM yields consistently lower scores, indicating superior predictive performance. However, because the models differ primarily in the tail region, the magnitude of the score differences is small.

Table 3.3: Energy score (ES) and threshold-weighted energy score (twES) for the BEMM and Gaussian models. Weighting scheme W_1 applies only to the region where all components exceed the 90th percentile of $\boldsymbol{E}$; W_2 uses the CDF of $\mathrm{MVN}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$ as the weighting function, with $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$ estimated from $\boldsymbol{E}$.

	ES	**twES** v_1	**twES** v_2
BEMM	2.0155	0.2512	1.0361
Gaussian	2.0197	0.2517	1.0384

3.6 Discussion

We now briefly discuss several practical challenges that may arise when implementing our model, along with potential solutions.

3.6.1 Other distributions for the bulk and alternative representations for the tail

The first issue concerns the choice of bulk and tail distributions. For the bulk distribution, we illustrate our framework using a multivariate normal distribution for simplicity; however, this is by no means the only viable option. Let $f_j(x_j)$ and $F_j(x_j)$, $j = 1, \dots, d$ denote the marginal density and CDF, respectively, and let $C(\cdot)$ be a copula with corresponding copula density $C(\cdot)$. A multivariate distribution $f(\boldsymbol{x})$ can be constructed as

$$f(\boldsymbol{x}) = c(F_1(x_1), \dots, F_d(x_d)) \prod_{i=1}^{d} f_j(x_j),$$

with its CDF F given by

$$F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)).$$

This allows for separate and flexible specification of the marginal distributions and the copula-based dependence structure to best fit the data. The multivariate normal distribution fits naturally within this copula framework, with copula C_R defined as

$$C_R(u_1, \dots, u_d) = \Phi_R(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_d)), \quad u_j \in [0, 1],$$

where Φ_R is the CDF of a d-dimensional standard normal with correlation matrix R.

For mGPD in the tail, Kiriliouk et al. (2019) provided several alternative forms of generator

f_U in (3.8), including those with independent components of Gumbel, reverse Gumbel, log-gamma, and multivariate normal components. These generators can be used in both R and T representations of the mGPD, leading to richer model forms. Recent research explores the use of generative models from machine learning to reformulate the mGPD (Hu and Castro-Camilo, 2025; Lhaut et al., 2025). Although these generative-model-based mGPDs can be integrated into our framework, the inference procedure would need to shift to a frequentist approach because the large number of parameters in generative models renders posterior sampling via MCMC computationally infeasible.

3.6.2 Model selection

Given the variety of possible model components, a robust model selection procedure is essential. Due to the large number of candidate models, information-criterion-based methods (e.g., WAIC (Watanabe and Opper, 2010)) are generally preferred. However, from a practical standpoint, fitting the full model in (3.11) across all combinations of bulk and tail distributions is computationally prohibitive, due to the large number of possible distribution combinations and the long MCMC runs required to obtain sufficient effective sample sizes in the presence of strong within-chain autocorrelation. We propose a heuristic two-stage approach for model selection:

1. **Stage One**: Since the bulk and tail distributions are independent for a fixed threshold, set the threshold to a relatively high quantile and select the best-fitting bulk and tail distributions separately, using criteria such as WAIC, on the bulk and tail data, respectively. This process can be repeated for multiple thresholds to account for variability, and the most frequently selected bulk and tail distributions are chosen.

2. **Stage Two**: Fit the model in (3.11) using the bulk and tail components identified in Stage One, following the inference procedure described in Section 3.3.4.

This approach avoids exhaustive MCMC fitting for all model combinations while still yielding a reasonable and practical model choice.

3.6.3 High-dimensional generalisation

Our model is best suited for low-dimensional settings (e.g. $d \leq 3$). Although it is theoretically applicable in higher dimensions, achieving satisfactory performance becomes challenging for two main reasons. First, in high-dimensional settings, complex cross-dimensional dependence structures are generally difficult to capture using distributions with simple copula constructions. While more flexible approaches, such as vine copulas (Kurowicka and Joe, 2010) or normalising flows Papamakarios et al. (2021), can be used to model the bulk distribution, these methods are primarily designed for density estimation and do not yield closed-form cumulative distribution functions. As a result, the exceedance probabilities required by our framework are difficult to evaluate. Second, the model's tail behaviour is fully determined by the mGPD, which is only justified when the data distribution is in the max-domain of attraction of an mGEVD. This is a strong condition that tends to hold in low dimensions but is rarely satisfied in high-dimensional contexts (Huser et al., 2025). In practice, it is advisable to respect these constraints and limit the application of the model to low-dimensional cases.

3.7 Conclusion

This paper introduces a Bayesian multivariate extreme value mixture model designed to jointly capture bulk and extreme observations. The proposed method models both the marginal distributions and the dependence structure simultaneously, and circumvents the threshold selection challenge in peaks-over-threshold modelling by treating the threshold as a learnable parameter. Bivariate simulation studies demonstrate that our model performs well across diverse tail behaviours when properly specified, and remains competitive in estimation accuracy compared to alternative bivariate approaches, even under model misspecification. An application to UK daily maximum air temperatures shows that the proposed model substantially improves the precision of residual tail estimates relative to a Gaussian benchmark.

We conclude by noting two limitations. First, the supports of the bulk and tail distributions may be misaligned. The support of the mGPD depends on the marginal shape parameter γ_j, and it could be lower bounded $\gamma_j > 0$ or upper bounded $\gamma_j < 0$. As a result, the bulk distribution may be unbounded below while the marginal mGPD has a lower bound, or the bulk distribution may be restricted to positive values while the mGPD extends to negative infinity. Since the lower tail of the mGPD generally carries little mass (see Figure 3.1), this mismatch is usually not problematic, but truncation of the mGPD can be applied if necessary. Second, because the mGPD contributes to both bulk and tail components, censored-likelihood-based inference cannot be applied to values below the threshold to reduce bias in parameter estimation (Kiriliouk et al., 2019). This means that mGPD estimation may remain biased even when the data exhibit asymptotic dependence. A potential remedy is to employ an mGPD with a flexible and learnable dependence structure, such as one based on normalising flows (Hu and Castro-Camilo, 2025), though this would require adapting the inference procedure to integrate deep learning techniques, as mentioned in Section 3.6.

Acknowledgements

We thank Lídia André (Lancaster University) for facilitating the code to implement the bivariate mixture copula model.

3.8 Supplementary materials

3.8.1 Code

Code to fit the bivariate version of our extreme mixture model and reproduce the data application in Section 3.5 is freely available at https://github.com/hcl516926907/Biv_Ext_Mix_Mod.

3.8.2 Proofs

We introduce the following Lemmas to facilitate the proof of Proposition 8

Lemma 1. *The mGPD $H(\boldsymbol{x})$ in (3.4) is in the MDA of the mGEVD $G(\boldsymbol{x})$.*

Proof. Theorem 2.2 in Rootzén and Tajvidi (2006) states that if $\boldsymbol{X}$ follows an mGPD $H(\boldsymbol{x})$ defined in (3.4), then there exists an increasing curve $\boldsymbol{u}(t)$ with $P(\boldsymbol{X} \leq \boldsymbol{u}(t)) \to 1$ as $t \to \infty$ and a function $\boldsymbol{\sigma}(\boldsymbol{u}) > \mathbf{0}$ such that

$$\mathbb{P}\left(\frac{\boldsymbol{X}-\boldsymbol{u}(t)}{\boldsymbol{\sigma}(\boldsymbol{u}(t))} \leq \boldsymbol{x} \,\middle|\, \frac{\boldsymbol{X}-\boldsymbol{u}(t)}{\boldsymbol{\sigma}(\boldsymbol{u}(t))} \nleq \mathbf{0}\right) = H(\boldsymbol{x}).$$

Following Theorem 2.1(ii) in the same paper, $H(\boldsymbol{x})$ is in the MDA of $G(\boldsymbol{x})$. □

Proof of Proposition 8. When all components exceed the threshold, we have

$$F(\boldsymbol{x}) = F_{\text{bulk}}(\boldsymbol{u}) + [1 - F_{\text{bulk}}(\boldsymbol{u})]H_{\boldsymbol{U}}(\boldsymbol{x}-\boldsymbol{u}), \quad \boldsymbol{x} > \boldsymbol{u}.$$

Let $\boldsymbol{x}_F^* = (x_1^*, \cdots, x_d^*)$, where $x_j^* = \sup\{F_j(x) < 1\}$ is the upper bound of margin i. Then,

$$\begin{aligned}\lim_{\boldsymbol{x}\to\boldsymbol{x}_F^{*-}} &= \frac{1 - H_{\boldsymbol{U}}(\boldsymbol{x}-\boldsymbol{u})}{1-F(\boldsymbol{x})}\\ &= \frac{1 - H_{\boldsymbol{U}}(\boldsymbol{x}-\boldsymbol{u})}{1 - F_{\text{bulk}}(\boldsymbol{u}) - [1 - F_{\text{bulk}}(\boldsymbol{u})]H_{\boldsymbol{U}}(\boldsymbol{x}-\boldsymbol{u})}\\ &= \frac{1}{1 - F_{\text{bulk}}(\boldsymbol{u})} > 0.\end{aligned}$$

By Lemma 1, there exist a sequence of vector $\boldsymbol{\alpha}_n > \boldsymbol{0}$ and $\boldsymbol{\beta}_n$ such that $H_{\boldsymbol{U}}^n(\boldsymbol{\alpha}_n(\boldsymbol{x}-\boldsymbol{u}) + \boldsymbol{\beta}_n) \to G(\boldsymbol{x})$. Following the proof to Theorem 2.1 in Resnick (1971), the above implies that $F^n(\boldsymbol{\alpha}_n\boldsymbol{x} + \boldsymbol{\beta}_n - \boldsymbol{\alpha}_n\boldsymbol{u}) \to G^{1-F_{\text{bulk}}(\boldsymbol{u})}(\boldsymbol{x})$, where $G^{1-F_{\text{bulk}}(\boldsymbol{u})}(\boldsymbol{x})$ is also an mGEVD by the max-stable property of G. □

Condition for Finite Expectation of BEMM. We restrict attention to the BEMM with a bivariate normal distribution for the bulk and an mGPD with an independent reverse exponential generator for the tail. Since the expectation of the bivariate normal is finite, it suffices to ensure that the mGPD component also has a finite expectation.

Let $\boldsymbol{Z} = (Z_1, Z_2)$ follow the standardized mGPD in (3.9). The mGPD random variable $\boldsymbol{X} = (X_1, X_2)$ is obtained from (3.7) as

$$\boldsymbol{X} = \frac{\boldsymbol{\sigma}}{\boldsymbol{\gamma}}(\exp\{\boldsymbol{\gamma}\boldsymbol{Z}\} - 1).$$

Then,

$$\mathbb{E}(X_j) = \mathbb{E}(X_j \mid X_j \geq 0)\mathbb{P}(X_j \geq 0) + \mathbb{E}(X_j \mid X_j < 0)\mathbb{P}(X_j < 0)$$

The first expectation term, $\mathbb{E}(X_j \mid X_j > 0)$, is the expectation of a univariate GPD, and hence is finite if $\gamma_j < 1$. The second term is

$$\mathbb{E}(X_j \mid X_j \leq 0) = \frac{\sigma_j}{\gamma_j}\mathbb{E}(\exp\{\gamma_j Z_j\} \mid Z_j \leq 0) - 1,$$

where

$$\begin{aligned}\mathbb{E}(\exp\{\gamma_j Z_j\} \mid Z_j \leq 0) &= \int_{-\infty}^{0} \exp\{\gamma_j z_j\} \int_{0}^{\infty} h_{\boldsymbol{U}}(z_j, z_{-j})\mathrm{d}z_{-j}\mathrm{d}z_j \\ &\propto \int_{-\infty}^{0} \exp\{\gamma_j z_j\} \cdot a_j^{-1} \exp\{a_j^{-1} z_j\}\mathrm{d}z_j \\ &\propto \int_{-\infty}^{0} \exp\{(\gamma_j + a_j^{-1})z_j\}\mathrm{d}z_j\end{aligned}$$

The last integration is finite if and only if $\gamma_j + a_j^{-1} > 0$. □

Proof of Theoretical χ of BEMM. This proof is based on the mGPD with an independent reverse exponential generator (density given in (3.9)). Let F_1 and F_2 be the marginal distributions in (3.11), and let $\boldsymbol{u} = (u_1, u_2)$ be a threshold vector satisfying $F_j(u_j) < 1$, $j = 1, 2$.

For $\max\{F_1(u_1), F_2(u_2)\} < r < 1$,

$$\begin{aligned}\mathbb{P}(F_j(X_j) > r) &= \mathbb{P}\left(F_{\text{bulk}}(u_1, u_2) + [1 - F_{\text{bulk}}(u_1, u_2)]H_j(X_j - u_j) > r\right) \\ &= \mathbb{P}\left(H_j(X_j - u_j) > \frac{r - F_{\text{bulk}}(u_1, u_2)}{1 - F_{\text{bulk}}(u_1, u_2)}\right),\end{aligned}$$

where H_j is the jth marginal of the mGPD in (3.9). Let $r^* = [r - F_{\text{bulk}}(u_1, u_2)]/[1 - F_{\text{bulk}}(u_1, u_2)]$. Then

$$\begin{aligned}\chi = \lim_{r^* \to 1^-} \chi(r^*) &= \frac{\mathbb{P}(H_1(X_1 - u_1) > r^*, H_2(X_2 - u_2) > r^*)}{1 - r^*} \\ &= 1 - \left(\frac{1 + a_{(1)}}{1 + a_{(2)}}\right)^{1 + a_{(2)}^{-1}} \frac{a_{(2)}}{a_{(1)}} \frac{a_1 a_2}{a_1 a_2 + a_1 + a_2},\end{aligned}$$

where $a_{(1)} = \min\{a_1, a_2\}$, $a_{(2)} = \max\{a_1, a_2\}$. This follows from the χ of the mGPD with an independent reverse exponential generator, as derived in Kiriliouk et al. (2019). □

3.8.3 Stationarity check of the temperature data

We apply (3.19) to remove the seasonal cycle and reduce autocorrelation in the daily air temperature data. Stationarity is assessed using the autocorrelation function (ACF) of the residuals at each station (Figure 3.8). The ACF values are close to zero for lags up to 32 days, indicating that seasonality and autocorrelation have been effectively removed.

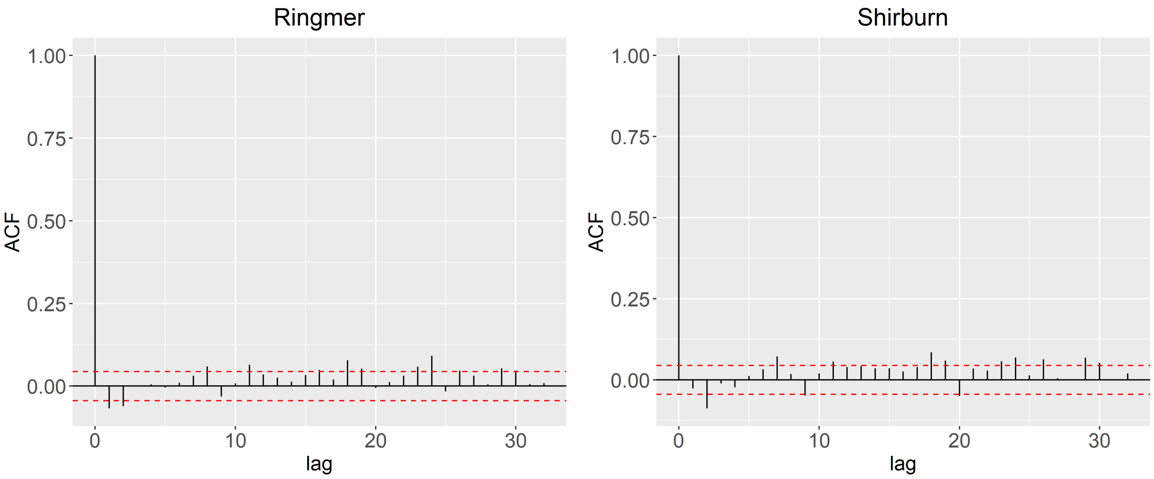

Figure 3.8: Autocorrelation plots of the two sites after removing the seasonality and autocorrelation by (3.19). The red dashed line indicates the confidence band at 95% confidence level.

3.8.4 MCMC convergence diagnostics

We assess convergence of the MCMC results, both in simulations and data applications, using the potential scale reduction factor $\hat{R}$ (Gelman and Rubin, 1992), the effective sample size (ESS), and trace plots. The statistic $\hat{R}$ compares the variance of parameter estimates across multiple chains (total variance) with the average variance within each chain. Well-mixed chains yield $\hat{R} \approx 1$. ESS, derived from $\hat{R}$, measures the number of effectively independent posterior draws. We use the rank-normalised $\hat{R}$ and ESS proposed by Vehtari et al. (2021), which are more robust than the traditional versions. For the data application, these diagnostics are shown in Table 3.1 and Fig. 3.1; simulation results are similar. Following Vehtari et al. (2021), stable inference typically requires $\hat{R} < 1.01$ and ESS > 400.

Table 3.1: Rank-normalised $\hat{R}$ and ESS (in parentheses) for the MCMC results in the data application. Three parallel chains of 20,000 iterations each were run. The first 10,000 iterations were discarded as burn-in, and the remaining samples were thinned by retaining every tenth draw.

Bulk Parameters		**Tail Parameters**		**Threshold**	
Param	**$\hat{R}$** (ESS)	**Param**	**$\hat{R}$** (ESS)	**Param**	**$\hat{R}$** (ESS)
$U[1,2]$	1.0007 (2034)	a_1	1.0015 (2101)		
$U[2,2]$	1.0007 (2030)	a_2	1.0009 (1992)		
μ_1	1.0044 (1478)	σ_1	1.0000 (1918)	u_1	1.0007 (775)
μ_2	1.0031 (1806)	σ_2	0.9996 (2662)	u_2	1.0018 (931)
s_1	1.0014 (1794)	γ_1	0.9997 (1670)		
s_2	1.0005 (1885)	γ_2	1.0006 (2716)		

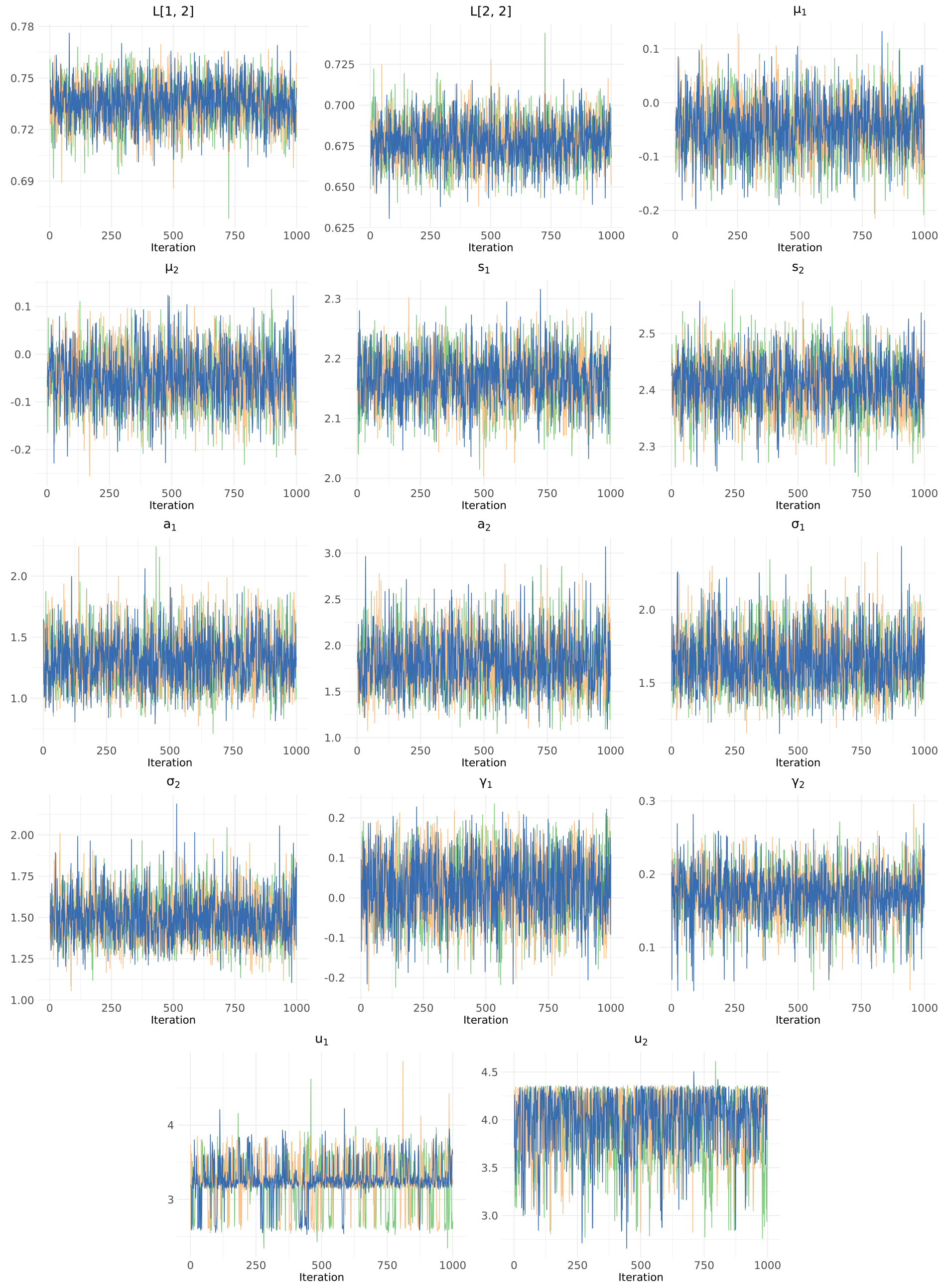

Figure 3.1: Trace plots for the MCMC results in the data application, with parameters ordered as in Figure 3.4.

Chapter 4

Multivariate GPD with normalising flows

The multivariate generalised Pareto distribution (mGPD) is a common method for modelling extreme threshold exceedance probabilities in environmental and financial risk management. Despite its broad applicability, mGPD faces challenges due to the infinite possible parametrisations of its dependence function, with only a few parametric models available in practice. To address this limitation, we introduce GPDFlow, an innovative mGPD model that leverages normalising flows to flexibly represent the dependence structure. Unlike traditional parametric mGPD approaches, GPDFlow does not impose explicit parametric assumptions on dependence, resulting in greater flexibility and enhanced performance. Additionally, GPDFlow allows direct inference of marginal parameters, providing insights into marginal tail behaviour. We derive tail dependence coefficients for GPDFlow, including a bivariate formulation, a d-dimensional extension, and an alternative measure for partial exceedance dependence. A general relationship between the bivariate tail dependence coefficient and the generative samples from normalising flows is discussed. Through simulations and a practical application analysing the risk among five major US banks, we demonstrate that GPDFlow significantly improves modelling accuracy and flexibility compared to traditional parametric methods.

4.1 Introduction

In multivariate extremes, accurately estimating joint exceedance probabilities is critical for risk management. Traditionally, this refers to the probability that all components exceed their respective thresholds simultaneously. However, recent studies increasingly focused on partial joint exceedance probabilities, where only a subset of the components surpasses thresholds (Heffernan and Tawn, 2004; Winter et al., 2016; Li et al., 2024).

Estimating these probabilities falls within the realm of multivariate threshold exceedance modelling, which is typically approached using the multivariate generalised Pareto distribution (mGPD). The mGPD naturally extends the univariate generalised Pareto distribution (GPD), a key concept in extreme value theory, as established by Balkema and De Haan (1974). Their seminal work led to the Pickands-Balkema-de Haan theorem, which demonstrates that, under certain conditions, the distribution of exceedances above a high threshold converges to a GPD. The class of distributions satisfying this theorem is linked to the class of distributions described by the Fisher–Tippett–Gnedenko theorem, thereby establishing a connection between the univariate GPD and the univariate generalised extreme value distribution (GEVD). This relationship was later extended to the multivariate case by Rootzén and Tajvidi (2006), who also demonstrated that the mGPD is the only threshold-stable multivariate distribution. While the univariate GPD has been widely applied in threshold exceedance models (Davison and Smith, 1990; Chavez-Demoulin and Davison, 2005; Castro-Camilo et al., 2019; He et al., 2022), its multivariate counterpart has seen relatively less use, primarily due to the challenges in extending the univariate formulation to the multivariate case. This is mainly due to the absence of a unique parametrisation, as the dependence structure in the multivariate GEVD allows for infinitely many possible parameterisations (Coles et al., 2001). To overcome this, Rootzén and Tajvidi (2006); Rootzén et al. (2018b,a) and Kiriliouk et al. (2019) proposed several parametrisations specifically designed to yield closed-form expressions that facilitate efficient computation. However, these parametrisations often fail to fully capture the complexity of real-world data.

To address this, we propose GPDFlow, an innovative flow-based mGPD model that employs normalising flows to flexibly represent the dependence structure. Normalising flows are powerful deep generative models that transform simple base distributions (such as Gaussian distributions) through a sequence of invertible mappings, enabling the approximation of highly complex distributions (Papamakarios et al., 2021). By leveraging normalising flows, GPDFlow avoids restrictive assumptions on tail dependencies, allowing the data to dictate dependence structures organically. Furthermore, embedding normalising flows within the mGPD framework makes GPDFlow statistically rigorous and maintains the desirable theoretical properties of traditional mGPD models.

The GPDFlow model offers two key advantages. First, it simultaneously captures marginal distributions and dependence structures within a unified modelling framework, significantly improving existing two-step extremal dependence modelling approaches that use generative approaches (Boulaguiem et al., 2022; McDonald et al., 2022). The marginal scale and shape parameters are explicitly estimated outside the flow transformations, providing direct insight into tail heaviness and overcoming known difficulties of affine triangular-based flow models in modelling heavy tails (Jaini et al., 2020). Second, GPDFlow serves as a proper statistical distribution with an explicit density function, facilitating straightforward maximum likelihood estimation and efficient sampling. Consequently, statistical inference, including extrapolation via Monte Carlo methods, is both reliable and computationally practical.

The rest of the paper is organised as follows. Section 4.2 provides the theoretical foundations of mGPD, normalising flows, and our proposed GPDFlow model. In Section 4.3, we present the inference procedure and derive an explicit formulation for the bivariate tail dependence coefficient of a GPDFlow. We further extend this analysis to the d-dimensional case ($d > 2$) and introduce an alternative coefficient designed to quantify the dependence of partial exceedances. Besides, we explore how the GPDFlow tail dependence coefficient relates to the broader properties of samples generated by normalising flows embedded within GPDFlow for $d = 2$. Simulation studies in Section 4.4 assess GPDFlow's performance under both correctly specified and misspecified conditions. Section 4.5 illustrates the practical utility of GPDFlow in financial risk management, specifically through calculating the Conditional Value at Risk (CoVaR) for negative returns of five major U.S. banks. Lastly, Section 4.6 acknowledges existing limitations of GPDFlow and outlines practical strategies to address these challenges.

Notation: Throughout the paper, $\max(\cdot)$ and $\min(\cdot)$ take one vector as input, returning its largest or smallest element, respectively, while $\wedge$ operates on two vectors, returning their elementwise minima. All operations between two vectors $\boldsymbol{x} = (x_1, \cdots, x_d)$ and $\boldsymbol{y} = (y_1, \cdots, y_d)$ are performed elementwise. For example, $\boldsymbol{xy}$ denotes the Hadamard product of $\boldsymbol{x}$ and $\boldsymbol{y}$, i.e. $\boldsymbol{xy} = (x_1y_1, \cdots, x_dy_d)$, and $\boldsymbol{x} \wedge \boldsymbol{y}$ represents the elementwise minima of $\boldsymbol{x}$ and $\boldsymbol{y}$, where $(\boldsymbol{x} \wedge \boldsymbol{y})_j = \min\{(x_j, y_j)\},\ j = 1, \cdots, d$. Similarly, logarithm, exponential, and indicator functions are applied elementwise when operating on vectors.

4.2 Method

4.2.1 Multivariate GPD

Suppose $\boldsymbol{Y}$ is a d-dimensional vector with cumulative distribution function F that is in the max-domain of attraction of a non-degenerated distribution G. This is to say, there exists sequences

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

$$\lim_{n\to\infty} n\{1 - F(\boldsymbol{a}_n \boldsymbol{y} + \boldsymbol{b}_n)\} = -\log G(\boldsymbol{y}). \tag{4.1}$$

The multivariate generalised Pareto distribution (mGPD) $H(\boldsymbol{x})$ arises when considering the conditional probability of $\boldsymbol{Y}$ when at least one component of $\boldsymbol{Y}$ is extreme (Rootzén and Tajvidi, 2006). Formally,

$$\begin{aligned} H(\boldsymbol{x}) &= \lim_{n\to\infty} \mathrm{P}\{\boldsymbol{a}_n^{-1}(\boldsymbol{Y} - \boldsymbol{b}_n) \leq \boldsymbol{x} | \boldsymbol{Y} \nleq \boldsymbol{b}_n\} \\ &= \frac{1}{\log G(\boldsymbol{0})} \log \frac{G(\boldsymbol{x} \wedge \boldsymbol{0})}{G(\boldsymbol{x})} \end{aligned} \tag{4.2}$$

The distribution G in (4.1) and (4.2) is the multivariate generalised extreme value distribution. As a direct result of the weak convergence in (4.1), which implies the convergence of both margin and copula, the marginal distributions $G_j(x_j)$, $j = 1, \dots, d$, are univariate generalised extreme value distributions by Fisher–Tippett–Gnedenko theorem (Haan and Ferreira, 2006, p. 6), with distribution function

$$G_j(x_j) = \begin{cases} \exp\left\{-\{1 + \gamma_j (x_j - \mu_j)/\alpha_j\}^{-1/\gamma_j}\right\} & \text{if } \gamma_j \neq 0, \\ \exp\{-\exp\{-(x_j - \mu_j)/\alpha_j\}\} & \text{if } \gamma_j = 0, \end{cases}$$

where $\alpha_j \in (0,\infty), \mu_j, \gamma_j \in \mathbb{R}$ and support $\{x_j \in \mathbb{R} : \alpha_j + \gamma_j(x_j - \mu_j) > 0\}$. Appropriate choices of $\boldsymbol{a}_n$ and $\boldsymbol{b}_n$ always ensure that 0 is in the support of G_j. The convergence of the corresponding copula implies

$$\lim_{n\to\infty} n\left\{1 - C_F\left(1 - \frac{x_1}{n}, \cdots, 1 - \frac{x_d}{n}\right)\right\} = -\log C_G\left(\exp\{-x_1\}, \cdots, \exp\{-x_d\}\right) := \ell(\boldsymbol{x})$$

where C_F and C_G are the copulas of F and G, respectively, and $\ell(\boldsymbol{x})$ is called the stable tail dependence function (stdf) of C_G, which describes the dependence structure of G (Segers, 2012). The stdf $\ell(\boldsymbol{x})$ does not have finite parametrisations. In fact, any function $\ell(\boldsymbol{x}) : [0,\infty)^d \to [0,\infty)$ is a valid stdf if it can be written as

$$\ell(\boldsymbol{x}) = \mathbb{E}[\max(\boldsymbol{x}\boldsymbol{V})], \tag{4.3}$$

where $\boldsymbol{V}$ is a d-dimensional random variable on $[0,\infty)^d$ satisfying $\mathbb{E}(V_j) = 1,\ j = 1, \cdots, d$ (Rootzén et al., 2018b). The expression in (4.3) defines $\ell(\boldsymbol{x})$ as a norm, call D-Norm, and $\boldsymbol{V}$ is the generator of $\ell(\boldsymbol{x})$ (Falk and Stupfler, 2017).

Combining the margins and the stdf, we can express $G(\boldsymbol{x})$ as

$$G(\boldsymbol{x}) = \exp\{-\ell\{-\log G_1(x_1), \cdots, -\log G_d(x_d)\}\}.$$

Consequently, $H(\boldsymbol{x})$ is determined by the marginal parameters $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_d), \boldsymbol{\mu} = (\mu_1, \cdots, \mu_d)$, $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_d)$ and the stdf $\ell(\boldsymbol{x})$. As noted by Rootzén et al. (2018b), the parametrization of H in terms of $(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\gamma}, \ell)$ is not convenient since $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}$ are not identifiable from H due to the max-stable property of G. This could be addressed by reparameterising H by $(\boldsymbol{\sigma}, \boldsymbol{\gamma}, \ell)$, where $\boldsymbol{\sigma} = \boldsymbol{\alpha} - \boldsymbol{\gamma}\boldsymbol{\mu}$ can be seen as the marginal scale parameter of $H(\boldsymbol{x})$. Note that $\sigma_j > 0$ is ensured by $G_j(0) > 0$.

If a random vector $\boldsymbol{X}$ follows a mGPD, then the marginal distributions

$$H_j(x_j) = \frac{1}{\log G(\boldsymbol{0})} \log \frac{G((0, \cdots, 0, x_j \wedge 0, 0, \cdots, 0))}{G_j(x_j)}, \quad \sigma_j + \gamma_j x_j > 0,\ j = 1, \cdots, d, \tag{4.4}$$

depend on the marginal parameters σ_j, γ_j and stdf ℓ. But the conditional distribution of $X_j > x_j | X_j > 0$ relies only on σ_j and γ_j:

$$\frac{1 - H_j(x_j)}{1 - H_j(0)} = \frac{\log G_j(x_j)}{\log G_j(0)} = \left(1 + \frac{\gamma_j x_j}{\sigma_j}\right)^{-1/\gamma_j}, \quad x_j > 0. \tag{4.5}$$

In other words, for an mGPD random vector, $\boldsymbol{X}$, the conditional distribution of $X_j > x_j | X_j > 0$ is a univariate GPD. Consequently, the parameters σ_j and γ_j retain the same interpretations as the scale and shape parameters of a univariate GPD. What's more, we can use the following transformation

$$\boldsymbol{Z} = g_{\text{std}}(\boldsymbol{X}; \boldsymbol{\sigma}, \boldsymbol{\gamma}) = \mathbb{1}\{\boldsymbol{\gamma} \neq \boldsymbol{0}\} \frac{1}{\boldsymbol{\gamma}} \log\left(\frac{\boldsymbol{\gamma}\boldsymbol{X}}{\boldsymbol{\sigma}} + 1\right) + \mathbb{1}\{\boldsymbol{\gamma} = \boldsymbol{0}\} \frac{\boldsymbol{X}}{\boldsymbol{\sigma}} \tag{4.6}$$

to standardize the margins so that $\boldsymbol{\sigma} = \boldsymbol{1}$, $\boldsymbol{\gamma} = \boldsymbol{0}$, and $\mathbb{P}(Z_j > z_j | Z_j > 0) = \exp(-z_j)$ for $z_j > 0$. We call this form the *standardised mGPD* and denote its cdf and density as $H(\boldsymbol{z})$ and $h(\boldsymbol{z})$, respectively.

Although $H(\boldsymbol{z})$ can be expressed as a function of $\boldsymbol{V}$ through (4.2), deriving $h(\boldsymbol{z})$ is less straightforward. Alternatively, $H(\boldsymbol{z})$ and $h(\boldsymbol{z})$ can be derived using a stochastic representation of the mGPD called the T representation (Rootzén et al., 2018b), given by

$$\boldsymbol{Z} = E + \boldsymbol{T} - \max(\boldsymbol{T}), \tag{4.7}$$

where E is a unit exponential random variable, and $\boldsymbol{T}$ is a d-dimensional random variable independent to E and satisfying the weak conditions $\mathrm{P}(T_j > -\infty) > 0$ and $\mathrm{P}(\max(\boldsymbol{T}) > -\infty) = 1$. $\boldsymbol{T}$ can be seen as a generator for the mGPD, which jointly influences the dependence and margins of $\boldsymbol{Z}$ with E. The generator $\boldsymbol{V}$ in $\ell(\boldsymbol{x})$ is associated with $\boldsymbol{T}$ by $\boldsymbol{V} = \exp\{\boldsymbol{T} - \max(\boldsymbol{T})\}/\mathbb{E}(\exp\{\boldsymbol{T} - \max(\boldsymbol{T})\})$. Using (4.7), the density function $h(\boldsymbol{z})$ is given by

$$h(\boldsymbol{z}) = \frac{\mathbb{1}\{\max(\boldsymbol{z} > 0)\}}{\exp\{\max(\boldsymbol{z})\}} \int_{-\infty}^{\infty} f_{\boldsymbol{T}}(\boldsymbol{z} + s)\mathrm{d}s, \tag{4.8}$$

where $f_{\boldsymbol{T}}$ is the density of $\boldsymbol{T}$. Since both conditions of $\boldsymbol{T}$ are satisfied by most common distributions, its flexibility allows for various parametrisations of the mGPD. For instance, Kiriliouk et al. (2019) studied the cases when $\boldsymbol{T}$ follows a multivariate Gaussian and several multivariate distributions with independent components (Gumbel, reverse Gumbel, reverse exponential, and log-gamma), in which closed-form expressions of $h(\boldsymbol{z})$ can be obtained.

There exist alternative representations of the mGPD based on point process representations of extreme episodes (Rootzén et al., 2018b,a). For instance, the R representation is derived directly from the point process in Section 3 of Rootzén et al. (2018a), while the U representation transforms R into a standardised scale by (4.6). Although both the R representation and U representation give rise to densities $h(\boldsymbol{z})$ slightly different from (4.8), they can be transformed into the T representation by adding constraints to the corresponding generators. Additionally, Theorem 4.4 in Rootzén et al. (2018b) highlighted that for any standardised mGPD, there always exists a random vector $\boldsymbol{T}$ capable of generating it. Therefore, without losing generality, we use the T representation here due to its simple density form and straightforward sampling.

4.2.2 Normalising flows

Normalising flows are flow-based generative models that apply a sequence of changes of variables to convert a simple distribution (base distribution) to any well-behaved distribution (target distribution). Starting with a d-dimensional random variable $\boldsymbol{U}$ with base density $f_{\boldsymbol{U}}(\boldsymbol{u})$ and a invertible and differentiable transformation g (namely a diffeomorphism), the transformed random variable $\boldsymbol{Y} = g(\boldsymbol{U})$ has density

$$f_{\boldsymbol{Y}}(\boldsymbol{y}) = f_{\boldsymbol{U}}(\boldsymbol{u}) |\det J_g(\boldsymbol{u})|^{-1}, \quad \boldsymbol{u} = g^{-1}(\boldsymbol{y}), \tag{4.9}$$

where $J_g(\boldsymbol{u})$ is the $d \times d$ Jacobian matrix of g, i.e.,

$$J_g(\mathbf{u}) = \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \cdots & \frac{\partial g_1}{\partial u_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_d}{\partial u_1} & \cdots & \frac{\partial g_d}{\partial u_d} \end{bmatrix}.$$

In the generative model context, (4.9) represents the generative direction, where "noise" from the base distribution is transformed into the target distribution. This process also corresponds to the sampling mechanism in a flow-based model. The reversed process of (4.9) is the normalising process, which transforms a target sample into the base sample and provides information for updating J_g. Such diffeomorphism g always exists for any well-behaved $f_{\boldsymbol{Y}}(\boldsymbol{y})$ and $f_{\boldsymbol{U}}(\boldsymbol{u})$ (Papamakarios et al., 2021). To construct a g that is expressive enough in the sense that it can be used to transform $f_{\boldsymbol{U}}$ to any $f_{\boldsymbol{Y}}$, we can first use a neural network to represent a diffeomorphism g_i, $i = 1, \ldots, K$, where its expressive power is justified by the universal approximation theorem (Hornik et al., 1989). We can then compose all g_i's into a single transformation $g = g_K \circ \cdots \circ g_1$ to further improve the expressivity of a single g_i.

A few architectures for g_i have been studied to ensure its differentiability and invertibility (Dinh et al., 2017; Kingma and Dhariwal, 2018; Papamakarios et al., 2017). In this paper, we use the real-valued non-volume preserving (Real NVP; Dinh et al. 2017), which constructs complex probability distributions by applying a sequence of invertible, affine transformations with a triangular Jacobian structure to a base distribution, enabling efficient density evaluation and sampling. Specifically, the Real NVP model constructs a transformation g by stacking multiple coupling layers together. In each coupling layer, an input vector $\boldsymbol{u}$ is partitioned into two parts $(\boldsymbol{u}_{p_1}, \boldsymbol{u}_{p_2})$. Then $\boldsymbol{u}_{p_1}$ remains unchanged and an affine transformation dependent on $\boldsymbol{u}_{p_1}$ is applied to $\boldsymbol{u}_{p_2}$. The mathematical formulation for such single coupling layer transformation $g_k : \boldsymbol{y} = g_k(\boldsymbol{u})$ is given by

$$\boldsymbol{y} = \boldsymbol{b}\boldsymbol{u} + (\boldsymbol{1} - \boldsymbol{b}) \left\{ \boldsymbol{u} \exp\left\{ \zeta_k(\boldsymbol{b}\boldsymbol{u}) \right\} + \upsilon_k(\boldsymbol{b}\boldsymbol{u}) \right\}, \tag{4.10}$$

where $\zeta_k, \upsilon_k : \mathbb{R}^d \to \mathbb{R}^d$ are, respectively, the log-scale and translation functions, defined as multilayer perceptions (MLPs). MLP is a feedforward neural network that extends regression by learning nonlinear transformations via weighted sums and activation functions, optimised through backpropagation (Rumelhart et al., 1986). The vector $\boldsymbol{b}$ is a d-dimensional binary mask vector that masks components, i.e. determine which components are $\boldsymbol{u}_{p_1}$. Let $M = \{j = 1, \cdots, d : b_j = 1\}$ denote the set of masked indices and $N = \{1, \cdots, d\} \backslash M$. Then, (4.10) implies

$$\begin{aligned} \boldsymbol{y}_{j\in M} &= \boldsymbol{u}_{j\in M}, \\ \boldsymbol{y}_{j\in N} &= \boldsymbol{u}_{j\in N} \exp\{\zeta_k(\boldsymbol{bu})\}_{j\in N} + \upsilon_k(\boldsymbol{bu})_{j\in N}. \end{aligned} \tag{4.11}$$

The components in the masked set M are invariant under the transformation, while the components in the unmasked set N are applied to an affine transformation with scale $\exp\{\zeta_k(\boldsymbol{bu})\}_{j\in N}$ and translation $\upsilon_k(\boldsymbol{bu})_{j\in N}$. Due to the masking, $\boldsymbol{bu}$ is a d-dimensional vector that has $u_j,\ j \in M$ at the j-th component and 0 elsewhere. Hence, the scale $\exp\{\zeta_k(\boldsymbol{bu})\}_{j\in N}$ and translation $\upsilon_k(\boldsymbol{bu})_{j\in N}$ are functions of $\boldsymbol{u}_{j\in M}$, and by updating these functions, Real NVP will learn the dependence information between $\boldsymbol{u}_{j\in M}$ and $\boldsymbol{u}_{j\in N}$. To ensure comprehensive learning of the dependence, $\boldsymbol{b}$ is usually flipped in every coupling layer (i.e., we swap M and N).

An appealing feature of (4.11) is that the determinant of the Jacobian of g_k's can be easily calculated by rearranging the Jacobian into a lower-triangle matrix:

$$|\det(g_k)| = \left| \det\left(\begin{bmatrix} \mathbb{I}_{|M|} & \boldsymbol{0} \\ \boldsymbol{L}_{|N|\times|M|}(\boldsymbol{u}_{j\in M}) & \operatorname{diag}(\exp\{\zeta_k(\boldsymbol{bu})\}_{j\in N}) \end{bmatrix} \right) \right| = \prod_{j\in N} \exp\left\{\zeta_k(\boldsymbol{bu})\right\}_j .$$

Since $|\det(g_k)|$ only depends on the output of ζ_k and is irrelevant to the specific structure of ζ_k and υ_k, we can define ζ_k and υ_k as complex as needed without worrying about the differentiability or invertibility of g_i.

Suppose we specify a K-layer transformation $g = g_K \circ \cdots \circ g_1$ with g_k defined as in (4.10). Let $\boldsymbol{\theta}$ denote all weights in ζ_k and υ_k, $k = 1, \cdots, K$. Then, the estimation of a Real NVP model boils down to finding $\widehat{\boldsymbol{\theta}}$ that maximises the likelihood $f_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{\theta})$ given a known $f_{\boldsymbol{U}}(\boldsymbol{u})$. Gradient descent and backpropagation are usually used to update $\boldsymbol{\theta}$ and find the maximum likelihood. Due to the flexibility of g, $f_{\boldsymbol{U}}(\boldsymbol{u})$ does not have a huge influence on the expressivity of $f_{\boldsymbol{Y}}(\boldsymbol{y})$ in most situations, so $f_{\boldsymbol{U}}(\boldsymbol{u})$ is commonly set as a standard multivariate Gaussian (Dinh et al., 2017). The above reveals that the RealNVP model can provide an exact density of $f_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{\theta})$, and estimating a RealNVP model is much like fitting a statistical distribution. This exact likelihood inference is one of the main distinctions of normalising flows from other generative models.

4.2.3 GPDFlow

We propose a flow-based mGPD called GPDFlow by expressing $f_{\boldsymbol{T}}$ in (4.8) using Real NVP. Specifically, let $\boldsymbol{x}$ denote a d-dimensional threshold exceedance observation, and $\boldsymbol{z}$ be a standardized version of $\boldsymbol{x}$, i.e. $\boldsymbol{z} = g_{\text{std}}(\boldsymbol{x}; \boldsymbol{\sigma}, \boldsymbol{\gamma})$ as defined in (4.6), GPDFlow is a d-dimensional distribution with density function

$$f(\boldsymbol{x}; \boldsymbol{\sigma}, \boldsymbol{\gamma}, \boldsymbol{\theta}) = \frac{\mathbb{1}\{\max(\boldsymbol{z} > 0)\}}{\exp\{\max(\boldsymbol{z})\}} \int_{-\infty}^{\infty} f_{\boldsymbol{U}}\{g^{-1}(\boldsymbol{z} + s; \boldsymbol{\theta})\} |\det J_g(\boldsymbol{z} + s; \boldsymbol{\theta})|^{-1} \mathrm{d}s \prod_{j=1}^{d} \frac{1}{\sigma_j + \gamma_j x_j}, \tag{4.12}$$

where $f_{\boldsymbol{U}}(\boldsymbol{u})$ is the density of a d-dimensional standard Gaussian distribution, $\boldsymbol{\sigma}$, $\boldsymbol{\gamma}$ are marginal parameters, g is the entire transformation in normalising flows with $\boldsymbol{\theta}$ as the total weights.

Equation (4.12) can be understood in two ways. From a statistical perspective, we are parametrising $f_{\boldsymbol{T}}$ in (4.8) by normalising flows; hence, GPDFlow is a valid mGPD and inherits all useful properties. These include threshold stability, closure under conditioning and marginalisation (i.e., lower-dimensional conditional margins remain mGPD), and sum-stability under appropriate shape constraints: for any nonnegative weight vector, the conditional distribution of a weighted sum of components, given that the weighted sum is positive, follows a univariate GPD (Kiriliouk et al., 2019). Viewing (4.12) from a normalising flows angle, a random variable $\boldsymbol{X}$ with density f can be obtained by

$$\boldsymbol{X} = g_{\text{std}} \circ g_{\text{mGPD}} \circ g(\boldsymbol{U}), \tag{4.13}$$

where $g_{\text{mGPD}}(\boldsymbol{T}) = E + \boldsymbol{T} - \max(\boldsymbol{T}),\ E \sim \exp(1)$. For the output $g(\boldsymbol{U})$ from Real NVP, the transformation g_{mGPD} squeezes it into a reversed L-shape (see Figure 4.3), ensuring that the output satisfies the mGPD's properties. The final layer transformation g_{std} adjusts for the scale of the observations via $\boldsymbol{\sigma}$ and accounts for marginal tail heaviness through $\boldsymbol{\gamma}$. A key advantage of the above structure is that it prevents Real NVP from directly modelling the marginal tail, where standard Real NVP often struggles to accurately estimate tail heaviness (Jaini et al., 2020).

Unlike recent two-step generative approaches in extreme value modelling, which first transform margins to a standard scale before estimating the dependence (Boulaguiem et al., 2022), GPDFlow jointly estimates both the marginal distribution and tail dependence for threshold exceedances. The marginal distribution of the mGPD in (4.4) depends on ℓ and, consequently, on $\boldsymbol{T}$. As a result, the flexibility of $f_{\boldsymbol{T}}$ in GPDFlow helps reduce the bias in marginal estimations compared to the classic mGPD, which often struggles to balance marginal and tail dependence estimation. Traditional mGPD models typically rely on the censored likelihood method to alleviate the bias introduced by an inappropriate description of the lower tail margins (Kiriliouk et al., 2019). However, this approach can become computationally expensive in high dimensions.

4.3 Inference

4.3.1 Likelihood and model estimation

Given d-dimensional observations $\boldsymbol{x}_i = (x_{i1}, \cdots, x_{id})$, $i = 1, \cdots, N$, we estimate the parameter $(\boldsymbol{\sigma}, \boldsymbol{\gamma}, \boldsymbol{\theta})$ in GPDFlow by maximizing the full log-likelihood

$$\begin{aligned} l(\boldsymbol{\sigma}, \boldsymbol{\gamma}, \boldsymbol{\theta}) = \sum_{i=1}^{N} \Bigg\{ \log \left\{ \int_{-\infty}^{\infty} f_{\boldsymbol{U}}\{g^{-1}(\boldsymbol{z}_i + s; \boldsymbol{\theta})\} |\det J_g(\boldsymbol{z}_i + s; \boldsymbol{\theta})|^{-1} \mathrm{d}s \right\} \\ + \log \mathbb{1}\{\max(\boldsymbol{z}_i > 0)\} - \max(\boldsymbol{z}_i) - \sum_{j=1}^{d} \log(|\sigma_j + \gamma_j x_{ij}|) \Bigg\}, \quad \boldsymbol{z}_i = g_{\text{std}}(\boldsymbol{x}_i). \end{aligned} \tag{4.14}$$

The integrand in (4.14), which is the flow density, maps $\mathbb{R}^d$ to $\mathbb{R}$. However, the integral is univariate since the scalar s is added to each component of $\boldsymbol{z}_i$ and the integration is over s. As a result, the computation complexity does not scale with the dimension, and the integral can be effectively approximated using numerical methods such as the trapezoidal rule or Monte Carlo. In the practical implementation, $\boldsymbol{z}_i$ is expressed as

$$\boldsymbol{z}_i = \mathbb{1}\{\boldsymbol{\gamma} \neq \boldsymbol{0}\}\boldsymbol{\gamma}^{-1} \log(|\boldsymbol{\gamma}\boldsymbol{x}_i/\boldsymbol{\sigma} + 1|) + \mathbb{1}\{\boldsymbol{\gamma} = \boldsymbol{0}\}\boldsymbol{x}_i/\boldsymbol{\sigma}$$

to avoid taking the logarithm of a negative value. Additionally, an extra penalty term

$$\lambda \sum_i \sum_j \mathbb{1}\{\sigma_j + \gamma_j x_{ij} \leq 0\}(\sigma_j + \gamma_j x_{ij})^2$$

is added to (4.14) to penalize values of σ_j and γ_j that do not satisfy $\sigma_j + \gamma_j x_j > 0$. Here, λ is a hyperparameter that controls the penalty and is typically set to a large value.

An interesting characteristic of (4.8) is that $f_{\boldsymbol{T}}$ cannot be uniquely identified from $h(\boldsymbol{z})$. This could be straightforwardly seen from (4.7) by noticing that, for any d-dimensional random vector $\boldsymbol{R} = (R, \cdots, R)$, $\boldsymbol{T}$ and $\boldsymbol{T} + \boldsymbol{R}$ will always lead to same $\boldsymbol{Z}$. The expression, $\boldsymbol{T} - \max(\boldsymbol{T}) := \boldsymbol{S}$ is called spectral random vector in Rootzén et al. (2018b), and its density $f_{\boldsymbol{S}}$ can be identified from $h(\boldsymbol{z})$. A direct consequence of the unidentifiability of $f_{\boldsymbol{T}}$ is that $f_{\boldsymbol{T}}$ could have different estimated densities depending on the initial weights of Real NVP, an issue illustrated in Figure 4.1. The reason we still build GPDFlow on $f_{\boldsymbol{T}}$ rather than $f_{\boldsymbol{S}}$ is that $f_{\boldsymbol{T}}$ is an unbounded density, whereas the support of $f_{\boldsymbol{S}}$ is defined by the union of the coordinate hyperplanes where at least one coordinate is zero. Applying such constraints of support for $f_{\boldsymbol{S}}$ in normalising flows is difficult; by contrast, no constraints are required in the $f_{\boldsymbol{T}}$ modelling. Since we are only interested in $h(\boldsymbol{z})$ in the threshold exceedance modelling rather than specific $f_{\boldsymbol{T}}$, the unidentifiability of $f_{\boldsymbol{T}}$ is not an issue in the practical use of GPDFlow.

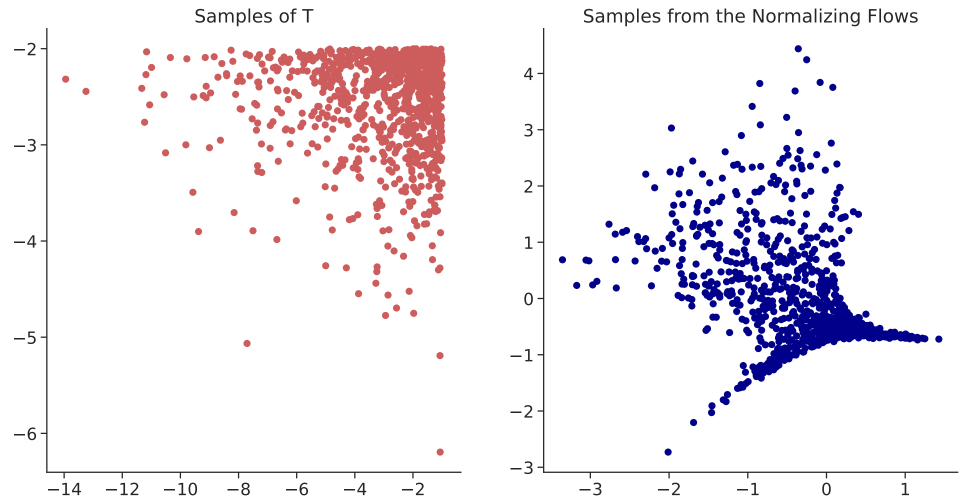

Figure 4.1: Comparison between 1000 samples (left) from $f_{\boldsymbol{T}}(t_1, t_2) = \exp\{(t_1+1)/2 + (t_2+2)/0.5\}$ $t_1 < -1, t_2 < -2$, and samples (right) from estimated $\widehat{f_{\boldsymbol{T}}}$ from the mGPD data generated by $f_{\boldsymbol{T}}$, illustrating the unidentifiability of $f_{\boldsymbol{T}}$.

Even though $f_{\boldsymbol{T}}$ is not identifiable, we aim to identify characteristics of the mGPD associated with $f_{\boldsymbol{T}}$, through which we can anticipate the behaviour of samples from the fitted normalising flows. The following proposition addresses this by examining the bivariate tail dependence coefficient of the mGPD.

Proposition 9. *Suppose F_1 and F_2 are the marginal cdfs of a bivariate mGPD with generator $\boldsymbol{T} = (T_1, T_2)$. Define q^* as the quantile level from which both components exceed the threshold, i.e., $q^* = \inf\{q \in (0,1) : F_1^{-1}(q) > 0$ and $F_2^{-1}(q) > 0\}$ and let $\chi_{1,2}(q) = \mathbb{P}(X_1 > F_1^{-1}(q) | X_2 > F_2^{-1}(q))$. Then, the bivariate tail dependence coefficient of the mGPD, $\chi_{1,2} = \lim_{q \to 1^-} \chi_{1,2}(q)$, can be expressed as*

$$\chi_{1,2} = \mathbb{E}\left(\min\left\{\frac{\exp\{T_1 - \max(\boldsymbol{T})\}}{\mathbb{E}(\exp\{T_1 - \max(\boldsymbol{T})\})}, \frac{\exp\{T_2 - \max(\boldsymbol{T})\}}{\mathbb{E}(\exp\{T_2 - \max(\boldsymbol{T})\})}\right\}\right). \tag{4.15}$$

Specifically,

$$\chi_{1,2}(q \mid q > q^*) = \chi_{1,2}.$$

If T_1 and T_2 are exchangeable, $\chi_{1,2}$ can be simplified to

$$\chi_{1,2} = \chi_{1,2}(q \mid q > q^*) = \frac{2\mathbb{E}(\exp\{-|T_1 - T_2|\})}{1 + \mathbb{E}(\exp\{-|T_1 - T_2|\})}.$$

Proposition 9 states that $\chi_{1,2}(q \mid q > q^*)$ is a positive constant and is equal to the bivariate tail dependence coefficient. When estimating GPDFlow, this constant line pattern of $\chi_{1,2}(q \mid q > q^*)$ is preserved regardless of the normalising flows output, and the value of $\chi_{1,2}(q \mid q > q^*)$ or $\chi_{1,2}$ determines the general behavior of $f_{\boldsymbol{T}}$. Under the exchangeability assumption, $\chi_{1,2}$ depends solely on the distribution of the difference between T_1 and T_2. If the data exhibit weak tail dependence (i.e., small $\chi_{1,2}$), T_1 and T_2 tend to spread across their sample spaces, exhibiting large differences. In contrast, strong tail dependence (large $\chi_{1,2}$) implies that T_1 and T_2 are more concentrated around similar values, and display smaller discrepancies. In the extreme case where $\chi_{1,2} = 1$, we have $T_1 = T_2$ almost surely. A proof for the proposition 9 can be found in Section 4.7.2 of the supplementary material.

4.3.2 Threshold selection

Finding a suitable threshold is crucial in driving accurate tail inference in threshold exceedance modelling, which involves bias and variance trade-offs in the estimation (Coles et al., 2001; Murphy et al., 2024). In the univariate setting, a common approach is to identify a quantity, such as the shape parameter ξ of the GPD, that remains constant across suitable thresholds and use a diagnostic plot to choose a threshold where this quantity stabilises. In the multivariate setting, we could exploit Proposition 9, which states that $\chi_{1,2}(q)$ remains constant for sufficiently large q. But instead of focusing on $\chi_{1,2}(q)$, we follow Kiriliouk et al. (2019) and generalise $\chi_{1,2}(q)$ to dimension $d > 2$. Similar to Proposition 9, let F_j be the marginal cdf of a d-dimensional mGPD, and $q^* = \inf\{q \in (0,1) : F_j^{-1}(q) > 0 \text{ for } \forall j = 1, \cdots, d\}$, define $\chi_{1:d}(q)$ as

$$\chi_{1:d}(q) = \frac{\mathbb{P}\{\bigcap_{j=1}^{d}\{X_j > F_j^{-1}(q)\}\}}{1-q},$$

and $\chi_{1:d} = \lim_{q \to 1^-} \chi_{1:d}(q)$, then for any $q^* < q < 1$, , we have

$$\chi_{1:d}(q \mid q > q^*) = \chi_{1:d}. \tag{4.16}$$

To determine an appropriate threshold, Kiriliouk et al. (2019) suggested calculating the empirical $\chi_{1:d}(q)$ from the data and using a diagnostic plot to define the threshold as the minimal marginal q-quantile such that $\widehat{\chi}_{1:d}(q) \approx \chi_{1:d}$ holds for $q > q^*$. There are two concerns with Kiriliouk's method. Firstly, $\widehat{\chi}_{1:d}(q)$ inevitably goes to zero even for a relatively small q in high dimensions due to sparsity in the joint exceedance region. Hence, this approach is less effective in high-dimensional applications. Secondly, $\widehat{\chi}_{1:d}(q)$ only considers the joint exceedance probability. When probabilities of partial threshold exceedances are of interest, $\chi_{1:d}(q)$ fails to evaluate the partial lower tail region, i.e., the region where some components fall below the threshold while others exceed it.

We therefore consider another quantity $\omega_{1:d}(q)$, defined as

$$\omega_{1:d}(q) = \frac{\mathbb{P}\{\bigcup_{j=1}^{d}\{X_j > F_j^{-1}(q)\}\}}{1-q}.$$

Let $\omega_{1:d} = \lim_{q\to 1^-} \omega_{1:d}(q)$ and q^* be the same as in (4.16), we have

$$\omega_{1:d}(q \mid q > q^*) = \omega_{1:d}.$$

Similar to $\chi_{1:d}(q)$ and $\chi_{1:d}$, $\omega_{1:d}(q)$ is constant when $q > q^*$ and hence equal to its limit $\omega_{1:d}$. The advantage of using $\omega_{1:d}(q)$ and $\omega_{1:d}$ is that they properly account for a suitable description of the lower tail margins of the mGPD, which is crucial if partial exceedances are of interest. Additionally, $\omega_{1:d}(q)$ accounts for all combinations of threshold exceedances, hence its empirical estimation is more stable than the empirical $\chi_{1:d}(q)$ due to the relatively large number of exceedances in the numerator, especially when the tail dependence of the data is weak. The following proposition summarises the above results and further establishes the form of $\chi_{1:d}$ and $\omega_{1:d}$ for an mGPD with random vector $\boldsymbol{T}$ as the generator. Proofs are deferred to Section 4.7.2 of the supplementary material.

Proposition 10. *Let F_j be the cdf of an mGPD, $q^* = \inf\{q \in (0,1) : F_j^{-1}(q) > 0 \, for \, \forall j = 1, \cdots, d\}$, then the following holds for $\chi_{1:d}$, $\chi_{1:d}(q)$, $\omega_{1:d}$, $\omega_{1:d}(q)$ defined previously:*

$$\begin{aligned}\chi_{1:d}(q \mid q > q^*) = \chi_{1:d} = \mathbb{E}\left(\min\left\{\frac{\exp\{T_1 - \max(\boldsymbol{T})\}}{\mathbb{E}(\exp\{T_1 - \max(\boldsymbol{T})\})}, \cdots, \frac{\exp\{T_d - \max(\boldsymbol{T})\}}{\mathbb{E}(\exp\{T_d - \max(\boldsymbol{T})\})}\right\}\right),\\ \omega_{1:d}(q \mid q > q^*) = \omega_{1:d} = \mathbb{E}\left(\max\left\{\frac{\exp\{T_1 - \max(\boldsymbol{T})\}}{\mathbb{E}(\exp\{T_1 - \max(\boldsymbol{T})\})}, \cdots, \frac{\exp\{T_d - \max(\boldsymbol{T})\}}{\mathbb{E}(\exp\{T_d - \max(\boldsymbol{T})\})}\right\}\right).\end{aligned} \tag{4.17}$$

In practice, one can use both $\chi_{1:d}(q)$ and $\omega_{1:d}(q)$ to find two appropriate thresholds and take the component-wise maxima as the final threshold. This approach is sketched in Algorithm (6).

Algorithm 6 Threshold Selection Method for GPDFlow

```
Input: All raw observations (rather than only the threshold-exceedance subset)
        $\boldsymbol{y}_i = (y_{i1}, \cdots, y_{id}), \quad i = 1, \cdots, N$
1. Calculate the empirical marginal cdf $\hat{F}_j(y_j) = \sum_{i=1}^N \mathbb{1}\{y_{ij} < y_j\}/N$ for $j = 1, \cdots, d$
2. For $q \in (0,1)$, calculate the empirical value of
    $\widehat{\chi}_{1:d}(q) = \frac{\sum_{i=1}^N \mathbb{1}\{\bigcap_{j=1}^d \{\widehat{F}_j(Y_j)>q\}\}}{N(1-q)}$
    $\widehat{\omega}_{1:d}(q) = \frac{\sum_{i=1}^N \mathbb{1}\{\bigcup_{j=1}^d \{\widehat{F}_j(Y_j)>q\}\}}{N(1-q)}$
3. Find the minimal $q_\chi$ such that $\widehat{\chi}_{1:d}(q)$ is approximately constant for $q > q_\chi$
4. Find the minimal $q_\omega$ such that $\widehat{\omega}_{1:d}(q)$ is approximately constant for $q > q_\omega$
5. Set $q^* = \max\{(q_\chi, q_\omega)\}$. The threshold is given by $(\hat{F}_1^{-1}(q^*), \cdots, \hat{F}_d^{-1}(q^*))$
Output: $d$-dimensional threshold vector $(\hat{F}_1^{-1}(q^*), \cdots, \hat{F}_d^{-1}(q^*))$
```

4.3.3 Hyperparameter tuning

The hyperparameters in GPDFlow correspond to those of the Real NVP architecture, including the number of transformations, the number of hidden layers, and the number of hidden neurons per layer in the MLPs used to construct the transformations. We tune these hyperparameters using standard hold-out validation, with 80% of the data randomly selected for training and the remaining 20% used for validation. Given the small sample size of the threshold exceedance data, we train the model for a relatively large number of epochs (200) and mitigate overfitting by monitoring the negative log-likelihood on the validation set, terminating training if no improvement is observed for 50 consecutive epochs. The hyperparameter configuration that achieves the minimum validation negative log-likelihood is selected as the final setting, and the model with the chosen hyperparameters and early-stopped epoch is refit to the full dataset for inference.

Throughout the simulation studies and data application, we fix the number of hidden layers in the MLPs to one and perform a grid search over $(4, 8, 12, 16)$ for the number of transformations and $(2d, 4d, 6d, 8d)$ for the number of hidden neurons in each MLP hidden layer. The corresponding results, including implementation details and learning curves, are reported in Section 4.7.3.

4.4 Simulation

We design two simulation scenarios to assess the estimation of GPDFlow. The first one is a well-specified scenario, where data are simulated from a traditional parametric mGPD. The purpose is to evaluate if GPDFlow can be correctly estimated. Specifically, we assess dependence and marginal estimation by examining all pairwise bivariate tail dependence coefficient estimates and marginal parameters $\boldsymbol{\sigma}$ and $\boldsymbol{\gamma}$. The second scenario is misspecified to test the model's robustness, with data drawn from a distribution constructed using the copula approach. Here, we compare the estimated density of threshold exceedance data and the joint exceedance probability.

In the first scenario, we evaluate the goodness of fit of GPDFlow in dimensions $d = 2, 3, 5$. The data are simulated from a parametric mGPD with $\boldsymbol{\sigma} = (0.5, 1.2, 1, 1.5, 0.8)$, $\boldsymbol{\gamma} = (-0.1, 0.2, 0, 0.15, -0.05)$, and $f_{\boldsymbol{T}}(\boldsymbol{t}) = \prod_{j=1}^{d} \exp\{t_j + \beta_j\}/a_j$, where $\boldsymbol{a} = (2, 0.5, 1, 5, 1.5)$ and $\boldsymbol{\beta} = (1, 2, 3, 4, 5)$. For $d = 2$ or 3, the parameters are restricted to the first d elements of each vector. For each dimension, we generate 100 samples from the parametric mGPD above and fit a GPDFlow using the best flow architecture in Table 4.1 to assess robustness under sparse data conditions.

We repeat the experiment 100 times for each dimension, and Figure 4.2 displays a comparison between the theoretical values of pairwise tail dependence, marginal parameters, and their corresponding Monte Carlo estimates. The closed-form of the theoretical pairwise tail dependence coefficient of the given $f_{\boldsymbol{T}}$ in this simulation scenario can be found in the supporting materials of Kiriliouk et al. (2019). Overall, the estimation performance is satisfactory, with most theoretical values falling within the interquartile range of the GPDFlow estimates. The model effectively captures both strong tail dependence ($\chi \approx 0.72$) and weak tail dependence ($\chi \approx 0.35$) present in the same dataset ($d = 5$). Additionally, GPDFlow accurately estimates the values of γ_j, correctly identifying whether each margin has a heavy, Pareto-like tail ($\gamma_j > 0$), a short or bounded tail ($\gamma_j < 0$), or a light, exponential tail ($\gamma_j = 0$). Notably, the estimations remain robust across dimensions, as we do not observe increasing uncertainty or significant bias as d grows.

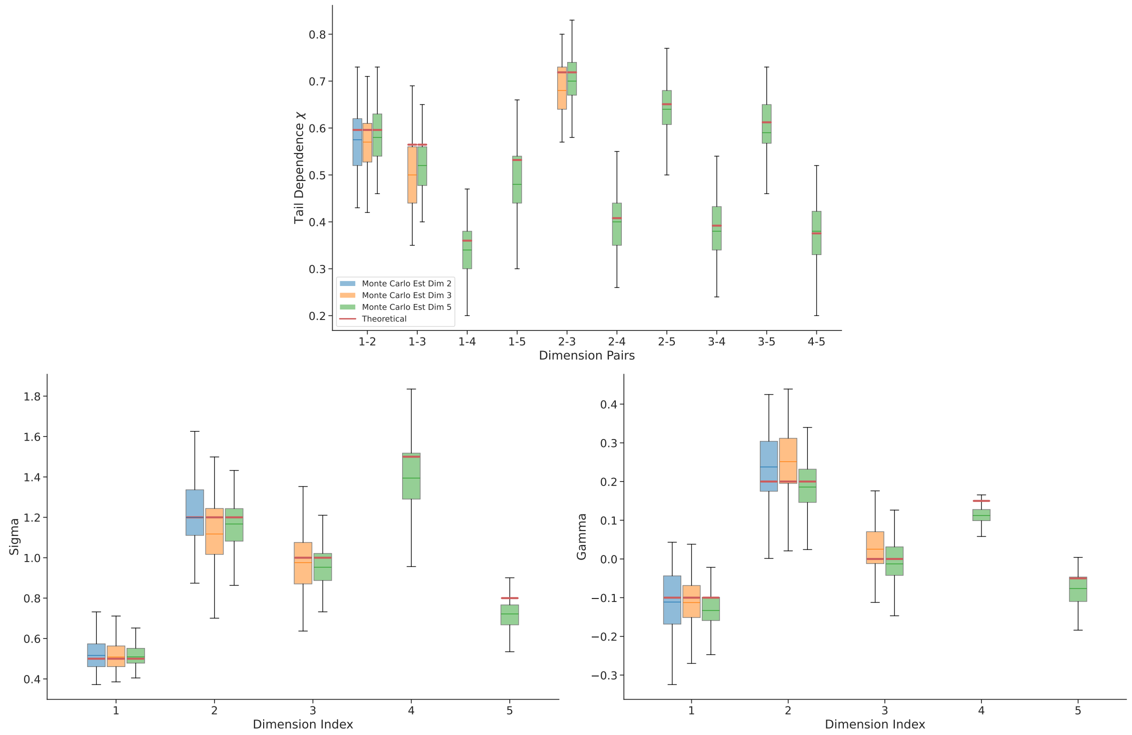

Figure 4.2: Boxplots of estimated pairwise χ (Top) and marginal parameters $\boldsymbol{\sigma}, \boldsymbol{\gamma}$ (Bottom two) across dimension $d = 2$, 3, and 5. The true model is an mGPD with $\boldsymbol{\sigma} = (0.5, 1.2, 1, 1.5, 0.8)$, $\boldsymbol{\gamma} = (-0.1, 0.2, 0, 0.15, -0.05)$, and $f_{\boldsymbol{T}}(\boldsymbol{t}) = \prod_{j=1}^{d} \exp\{t_j + \beta_j\}/a_j$, where $\boldsymbol{a} = (2, 0.5, 1, 5, 1.5)$ and $\boldsymbol{\beta} = (1, 2, 3, 4, 5)$. Point estimates for $\boldsymbol{\sigma}$ and $\boldsymbol{\gamma}$ are directly obtained from the GPDFlow estimation based on 100 simulations, while the point estimates for χ are based on the empirical values derived from 10,000 predictions in each of the 100 simulations.

In the second scenario, we assess GPDFlow's estimation performance when data are not in the mGPD framework. To facilitate visualisation, simulations are conducted in two dimensions. The data are generated from a bivariate Gumbel copula $C_{\text{Gumbel}}(\cdot; \theta)$ with parameter $\theta = 1.3$, and two Gaussian margins with location and scale parameters $(\mu_1, s_1) = (1, 3)$ and $(\mu_2, s_2) = (2, 5)$. In other words, the data comes from the distribution

$$F_{Y_1,Y_2}(y_1, y_2) = C_{\text{Gumbel}}\{\Phi[s_1^{-1}(y_1 - \mu_1)], \Phi[s_2^{-1}(y_2 - \mu_2)]; \theta\}, \tag{4.18}$$

where $C_{\text{Gumbel}}(u, v; \theta) = \exp\{-[(-\log u)^{\theta} + (-\log v)^{\theta}]^{1/\theta}\}$, and Φ is the cdf of a standard Gaussian distribution. We generate 1200 samples, and Algorithm 6 identifies $q = 0.95$ as a suitable threshold, yielding approximately 100 threshold exceedance samples. These exceedance samples are then fitted using a GPDFlow model with the best architecture presented in Table 4.2. To assess estimation accuracy, we repeat this procedure 100 times and compare the Monte Carlo mean of the GPDFlow estimates with the corresponding theoretical values.

The top two plots in Figure 4.3 compare the theoretical and GPDFlow-estimated densities of the threshold exceedance data. The theoretical density is obtained as

$$f_{X_1,X_2}(x_1,x_2) = \frac{f_{Y_1,Y_2}(x_1+\tau_1, x_2+\tau_2)}{1 - C_{\text{Gumbel}}(0.95, 0.95; \theta)} \mathbb{1}\left\{\bigcup_{j=1}^{2}\left\{\Phi\left(\frac{x_j+\tau_j-\mu_j}{s_j}\right) > 0.95\right\}\right\}, \tag{4.19}$$

where $\boldsymbol{\tau} = (\tau_1, \tau_2)$ is the threshold, and τ_j is the 0.95-quantile of margin j, $j = 1, 2$. (4.19) corresponds to a truncated $f_{Y_1,Y_2}(y_1, y_2)$ with support only in the region where at least one component exceeds the threshold. To facilitate the visualisation, we shift this support by subtracting the threshold and positioning the new boundary along negative x- and y-coordinates. The contour plots show that GPDFlow generally provides a good approximation of the theoretical density, particularly in high-density regions. Minor discrepancies appear in the lower tail and areas where the density is close to zero. These differences arise because GPDFlow defines a bounded distribution with support depending on $\boldsymbol{\gamma}$, whereas the theoretical density remains unbounded.

We further scrutinise the performance of GPDFlow by checking the estimation of joint probabilities associated with partial extremes. Specifically, we analize the estimation of $\mathbb{P}(Y_1 < q_{1,\alpha}, Y_2 > q_{2,0.99})$, where Y_1 and Y_2 are the random variables of the simulated data, $q_{1,\alpha}$ is the α-quantile of margin 1 and $q_{2,0.99}$ is the 0.99 quantile of margin 2. The theoretical value of such probability is straightforward to calculate by the cdf of the Gumbel Copula. Indeed, we have

$$\begin{aligned} \mathbb{P}(Y_1 < q_{1,\alpha}, Y_2 > q_{2,0.99}) &= \mathbb{P}(Y_1 < q_{1,\alpha}) - \mathbb{P}(Y_1 < q_{1,\alpha}, Y_2 < q_{2,0.99}) \\ &= \alpha - C_{\text{Gumbel}}(\alpha, 0.99; \theta). \end{aligned} \tag{4.20}$$

To estimate $\mathbb{P}(Y_1 < q_{1,\alpha}, Y_2 > q_{2,0.99})$ by GPDFlow, we can write

$$\begin{aligned} &\mathbb{P}(Y_1 < q_{1,\alpha}, Y_2 > q_{2,0.99}) \\ =&\mathbb{P}\left(Y_1 < q_{1,\alpha}, Y_2 > q_{2,0.99}, \bigcup_{j=1}^{2}\{Y_j > q_{j,0.95}\}\right) \\ =&\mathbb{P}\left(Y_1 - \tau_1 < q_{1,\alpha} - \tau_1, Y_2 - \tau_2 > q_{2,0.99} - \tau_2 \middle| \bigcup_{j=1}^{2}\{Y_j - \tau_j > 0\}\right) \mathbb{P}\left(\bigcup_{j=1}^{2}\{Y_j > \tau_j\}\right). \end{aligned} \tag{4.21}$$

The first term in the last equation of (4.21) is a probability conditioned on at least one extreme component, making it well-suited for estimation via GPDFlow. The second term is a threshold exceedance probability of a moderately high threshold, which can be well estimated empirically. The bottom plot in Figure 4.3 compared the theoretical probability $\mathbb{P}(Y_1 < q_{1,\alpha}, Y_2 > q_{2,0.99})$ with its Monte Carlo estimate from GPDFlow for $\alpha \in (0.5, 0.9)$ on a logarithm scale. The

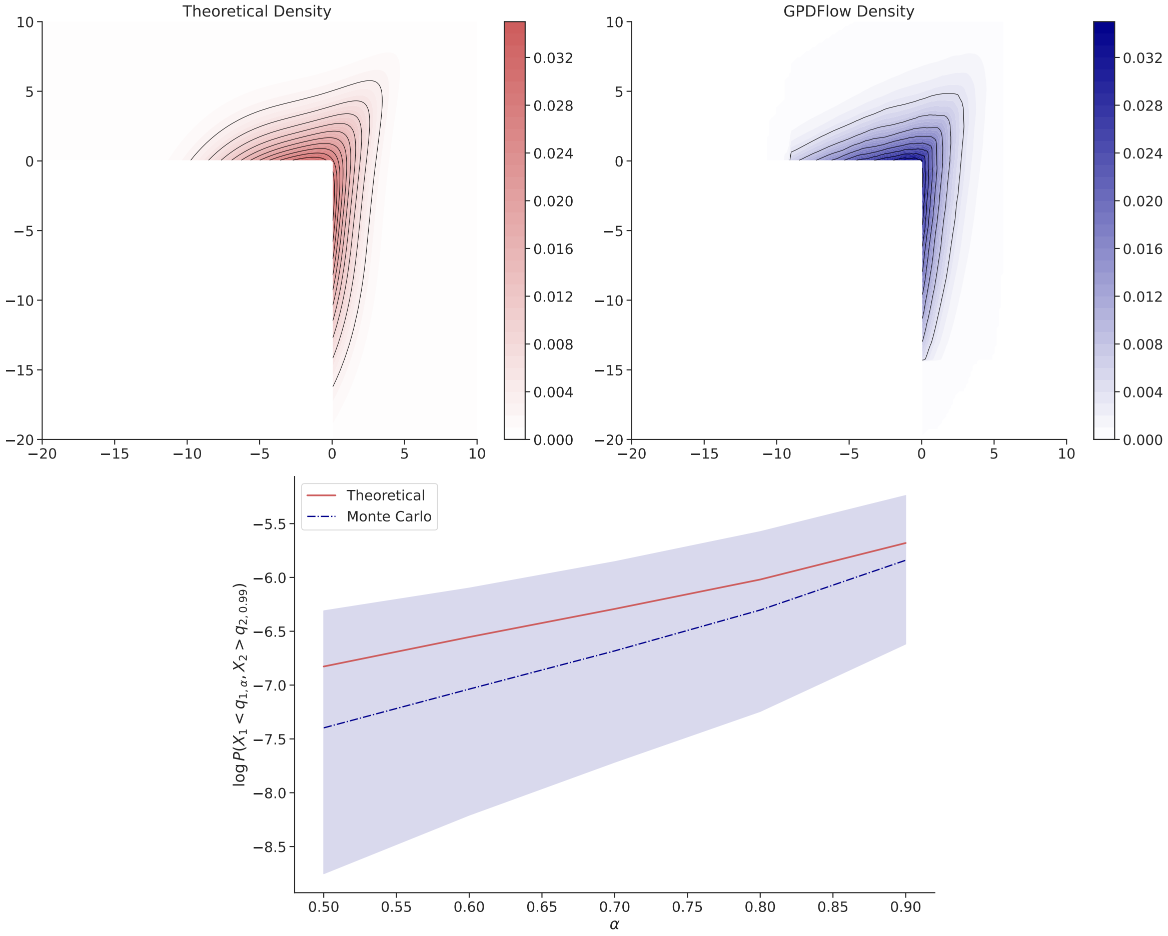

Figure 4.3: Theoretical and estimated density contour plots of the threshold exceedance data (top panel) and plot of $\log \mathbb{P}(Y_1 < q_{1,\alpha}, Y_2 > q_{2,0.99})$ (bottom). The entire data comes from the distribution in (4.18). The theoretical threshold density is obtained following (4.19), while the contour of the GPDFlow density is the average density over 100 simulations. In the bottom plot, the theoretical line is derived from (4.20), while the dark blue line is the mean of empirical estimation of (4.21) from 100,000 GPDFlow samples over 100 simulations, and the shaded area covers the 2.5% and 97.5% of the empirical estimations.

GPDFlow estimates closely align with the theoretical value, though a slight underestimation can be observed across all α. As α increases, both bias and variance decrease because the data become more extreme and thus more closely match the theoretical limiting distribution, allowing GPDFlow to characterise them more accurately.

4.5 Application

We apply our GPDFlow to multivariate risk analysis of five major US banks. The key risk measure in this study is Conditional Value-at-Risk (CoVaR), a pairwise Value-at-Risk that accounts for the dependence between two financial entities, such as institutions or portfolios. While the definition of CoVaR may vary, it generally falls within a framework that quantifies the risk of one entity given that one or more others are in distress. For instance, Mainik and Schaanning (2014) defines the CoVaR of two random variable X and Y as

$$\text{CoVaR}_{\alpha,\beta}(Y|X) = \text{VaR}_{\alpha}\{Y|X \geq \text{VaR}_{\beta}(X)\}, \tag{4.22}$$

where $\text{VaR}_{\eta}(Z) = \inf\{z \in \mathbb{R} : F_Z(z) \geq \eta\}$, $\eta \in (0,1)$, and F_Z is the cdf of random variable Z. This definition extends beyond individual risks by providing a method to measure systemic risk. Estimating CoVaR requires the entire marginal distribution conditioned on the extreme behaviour of the other component, a task that can be naturally handled by GPDFlow.

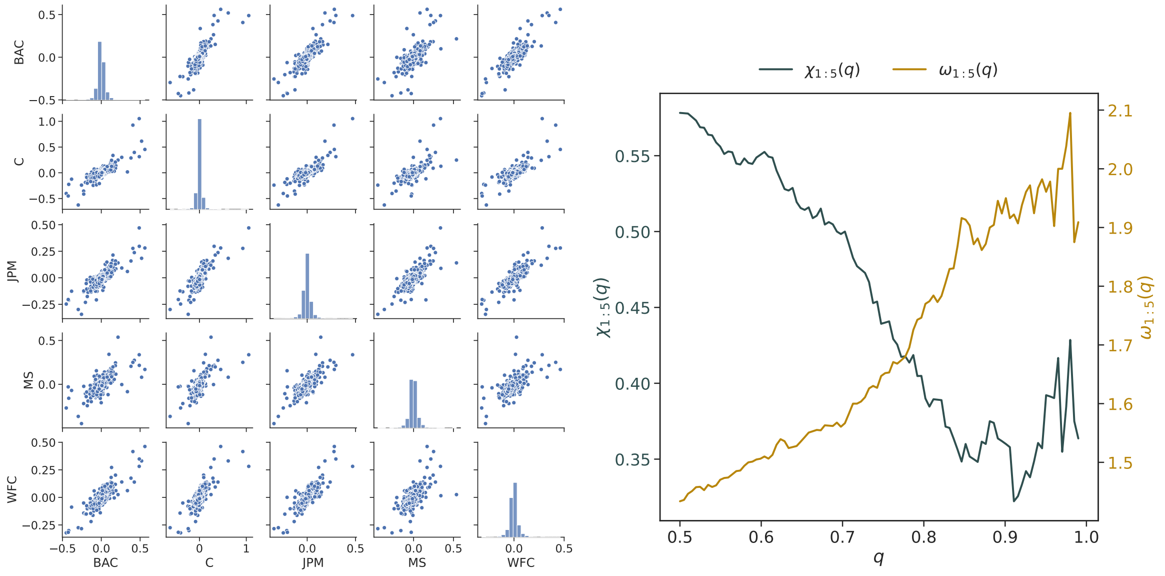

Figure 4.4: Scatter plots of the negative log-return (left) of the five US banks and empirical $\chi_{1:d}(q)$ and $\omega_{1:d}(q)$ of the negative log-return calculated by Algorithm 6.

We demonstrate the application of GPDFlow in risk management by examining the CoVaR of the following five large US banks: JPMorgan Chase (JPM), Bank of America (BAC), Citigroup (C), Wells Fargo (WFC) and Morgan Stanley (MS). Using 5-day closing prices from these banks between January 1, 2005, and February 1, 2025, sourced from Yahoo Finance, we examine the negative log returns, which total 1,010 observations. Since our focus is on general risk rather than forecasting, we do not remove heteroscedasticity from the negative log returns, which is the same treatment as in the analysis of the negative return of bank stocks in Kiriliouk et al. (2019). Figure 4.4 shows scatter plots of the negative log return as well as plots of the two tail dependence measures introduced in Proposition 10 for the five banks. Clear asymptotic dependence is observed in the scatter plot, which is confirmed by the positive values of $\widehat{\chi}_{1:5}(q)$ on the right. The empirical $\omega_{1:5}(q)$ seems to enter a plateau at around $q = 0.9$, while this happens at about $q = 0.95$ for $\chi_{1:5}(q)$. This suggests that the tail data can be well approximated by GPDFlow when at least one component exceeds its 0.95-quantile. Consequently, we select the marginal 0.95-quantile as the threshold and model the data above it (100 observations total) using GPDFlow.

For comparison purposes, we also implement a parametric mGPD from Kiriliouk et al. (2019) on our data and compare its performance with that of GPDFlow. This parametric mGPD is chosen from three mGPD models, where $f_{\boldsymbol{T}}$ is either independent reverse-exponential, independent Gumbel, or multivariate Gaussian, following the selection procedure in the original paper. The resulting model is given by $f_{\boldsymbol{T}}(t_1, \cdots, t_5) = \prod_{j=1}^{5} \alpha_j \exp(-\alpha_j t_j) \exp[-\exp(-\alpha_j t_j)],\ \alpha_j > 0$, with free parameters $\boldsymbol{\sigma}$ and $\boldsymbol{\gamma}$. Parameters in this model are estimated via a censored-likelihood method, in which components below the threshold are treated as censored.

Figure 4.5 compares the performance of GPDFlow and the parametric mGPD in estimating both the marginal densities and tail dependence of the five banks' threshold exceedance data. For the marginal density of JPMorgan Chase and Morgan Stanley (with others provided in Section 4.7.4 of the Appendix), GPDFlow provides a strong approximation across the lower tail, upper tail, and high-density regions, without overfitting to the wiggly empirical right tails. In contrast, although the parametric mGPD does a relatively good job of estimating the right tails, it significantly deviates from the empirical density in the lower-tail and high-density regions of the two banks.

In the tail dependence plots, both the estimated $\chi^m_{1:5}(q)$, $\omega^m_{1:5}(q)$, $m \in \{\text{GPDFlow, mGPD}\}$ from GPDFlow and the parametric mGPD show constant behavior for $q < 0.95$, consistent with Proposition 10. This constant trend theoretically extends to the region $q > 0.95$ but is subject to more volatility due to the limited number of predicted observations in the simulated data. Comparing the empirical $\chi_{1:5}$ and $\omega_{1:5}$ with the estimated $\chi^m_{1:5}$ and $\omega^m_{1:5}$ (approximately by $\chi^m_{1:5}(q \mid 0.5 < q < 0.95)$ and $\omega^m_{1:5}(q \mid 0.5 < q < 0.95)$, respectively), it is clear that the parametric mGPD underestimates $\chi_{1:5}$ and overestimates $\omega_{1:5}$. Its estimates lie near the boundary of the 95% bootstrap confidence interval of the empirical values. In contrast, GPDFlow estimations exhibit a much smaller bias, with both $\chi^{\text{GPDFlow}}_{1:5}$ and $\omega^{\text{GPDFlow}}_{1:5}$ falling well within the central region of the corresponding empirical confidence interval.

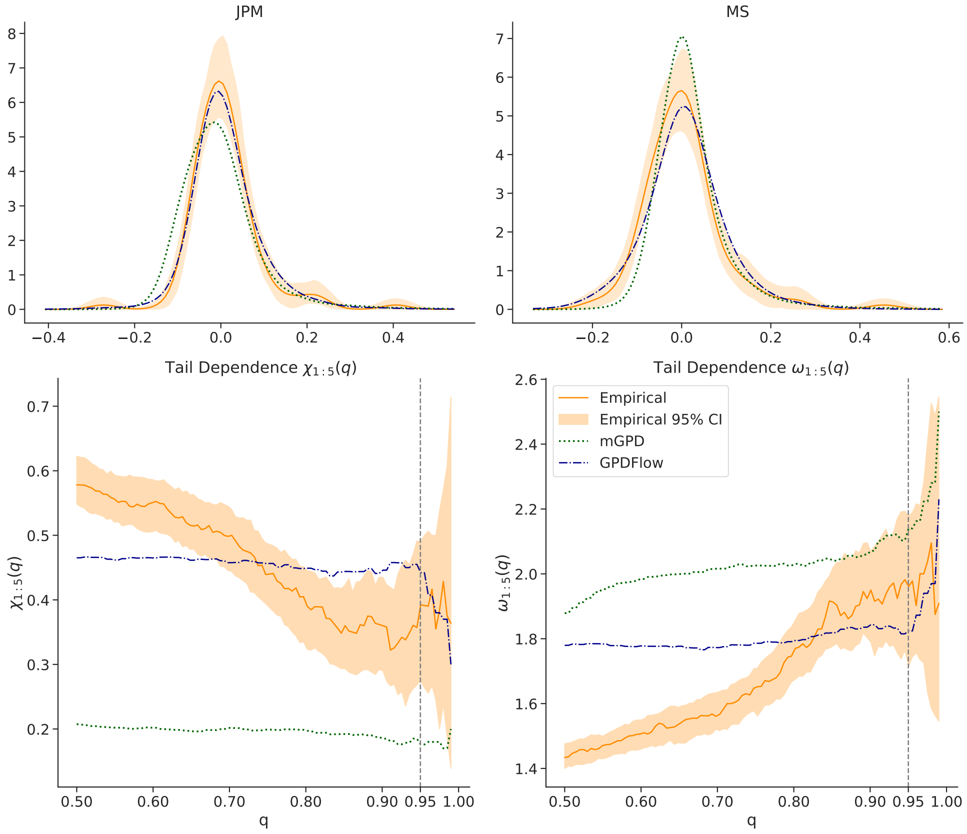

Figure 4.5: Plots of marginal densities of the threshold exceedance data (top panel) and tail dependence measures (bottom panel). The 95% CI for the empirical density and dependence measures is generated via Bootstrap. The estimates from the mGPD and GPDFlow are averages derived from 100 Monte Carlo sample sets, each of the same size as the threshold exceedance data and simulated using mGPD or GPDFlow.

Next, we use the GPDFlow predictions to estimate the CoVaR between the five banks. We illustrate the CoVaR that conditions on the largest bank, JPM, under a stress scenario (specifically, $\beta = 0.95$). Let the negative log-returns of the five banks be denoted by Y_{JPM}, Y_{BAC}, Y_{C}, Y_{WFC}, Y_{MS} and their corresponding 0.95-quantile threshold by τ_{JPM}, τ_{BAC}, τ_{C}, τ_{WFC}, τ_{MS}. By definition, for any two banks $i, j \in \{\text{BAC, C, JPM, MS, WFC}\},\ i \neq j$, (4.22) can be written as

$$\begin{aligned}\alpha &= \frac{\mathbb{P}\{Y_j < \text{CoVaR}_{\alpha,\beta}(Y_j|Y_i), Y_i > \text{VaR}_\beta(Y_i)\}}{\mathbb{P}\{Y_i > \text{VaR}_\beta(Y_i)\}} \\ &= \frac{\mathbb{P}\{Y_j < \text{CoVaR}_{\alpha,\beta}(Y_j|Y_i), Y_i > \text{VaR}_\beta(Y_i), \bigcup_{i=1}^5\{Y_i > \text{VaR}_\beta(Y_i)\}\}}{\mathbb{P}\{Y_i > \text{VaR}_\beta(Y_i), \bigcup_{i=1}^5\{Y_i > \text{VaR}_\beta(Y_i)\}\}} \\ &= \frac{\mathbb{P}\{Y_j < \text{CoVaR}_{\alpha,\beta}(Y_j|Y_i), Y_i > \text{VaR}_\beta(Y_i) | \bigcup_{i=1}^5\{Y_i > \text{VaR}_\beta(Y_i)\}\}}{\mathbb{P}\{Y_i > \text{VaR}_\beta(Y_i) | \bigcup_{i=1}^5\{Y_i > \text{VaR}_\beta(Y_i)\}\}}\end{aligned} \tag{4.23}$$

When $\beta \geq 0.95$, we have $\text{VaR}_\beta(Y_i) \geq \tau_i$. Consequently, the set $\bigcup_{i=1}^5\{Y_i > \text{VaR}_\beta(Y_i)\}$ is a subregion of the threshold exceedance data $\bigcup_{i=1}^5\{Y_i > \tau_i\}$, which Algorithm 6 indicates can be well approximated by GPDFlow. By setting a threshold $\boldsymbol{\tau}^*$, where $\boldsymbol{\tau}_i^* = \text{VaR}_\beta(Y_i)$, $i \in \{\text{BAC, C, JPM, MS, WFC}\}$, both the numerator and denominator in the last equation of (4.23) can be approximated by a GPDFlow model under this new threshold. Therefore, estimating $\text{CoVaR}_{\alpha,\beta}(Y_j|Y_i)$ on the raw negative log-return scale is equivalent to finding the $\text{CoVaR}_{\alpha,\beta}(Y_j - \tau_j^*|Y_i - \tau_i^*)$ on the threshold exceedance scale and then translating it back by τ_j^*. Figure 4.6 displays the CoVaR of BAC, C, MS and WFC conditioned on JMP being in distress (i.e., $Y_{\text{JPM}} > \text{VaR}_{0.95}(Y_{\text{JPM}})$). The GPDFlow estimates accurately fit the empirical CoVaR across all ranges of α, with larger uncertainty for higher α. Nevertheless, nearly all empirical CoVaR values fall within the 95% Monte Carlo confidence interval. Citigroup exhibits the strongest tail dependence with JPMorgan Chase, with $\text{CoVaR}_{\alpha,0.95}(\text{C}|\text{JPM})$ being higher for large α compared to the other three banks. Conversely, Morgan Stanley is less affected by JPMorgan Chase's distress, showing the smallest CoVaR among the four banks.

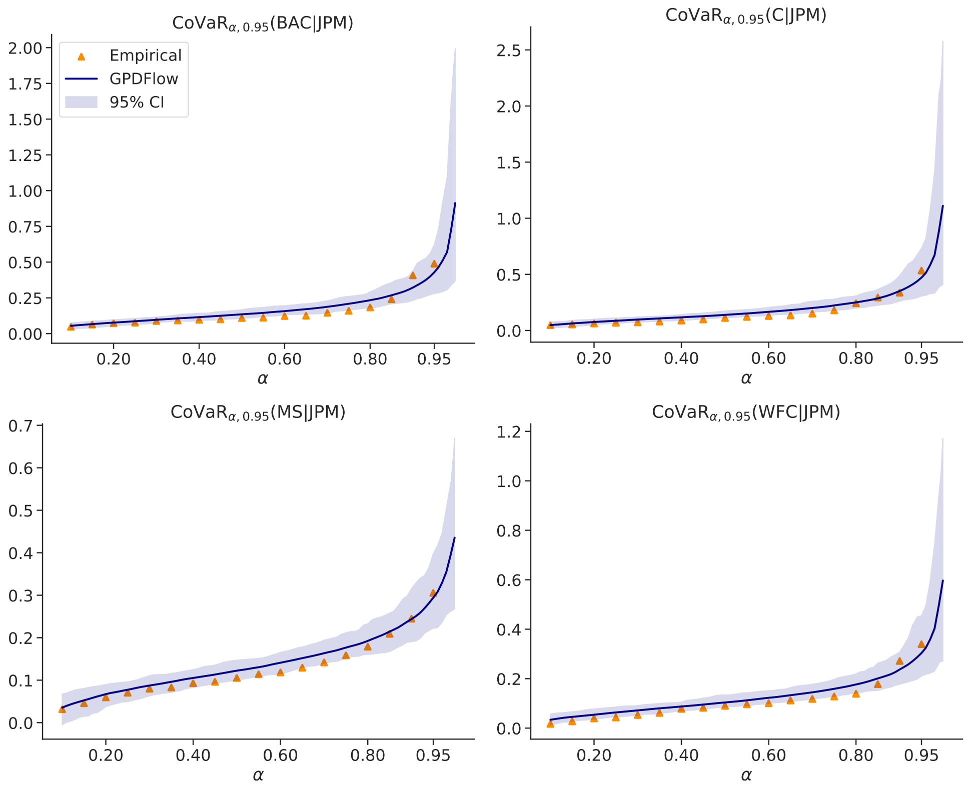

Figure 4.6: CoVaR$_{\alpha,\beta}(\cdot|\text{JPM})$ for the four banks, conditioned on JPM being in distress ($\beta = 0.95$). The GPDFlow line represents the average CoVaR from 100 Monte Carlo estimations, each derived from the GPDFlow simulated samples that are of the same size as the exceedance data. The 95% CI is derived from the 2.5% and 97.5% quantiles of the Monte Carlo estimations.

4.6 Discussion

In this paper, we introduce GPDFlow, a flow-based mGPD model that leverages normalising flows to capture dependence structures in multivariate threshold exceedances. GPDFlow explicitly estimates marginal parameters, allowing direct inference on tail behaviour while offering flexible dependence modelling. Simulation studies and a real data application show that GPDFlow achieves strong approximation performance for stationary threshold exceedance data.

The GPDFlow framework easily accommodates non-stationary extensions. Specifically, the marginal parameters $\boldsymbol{\sigma}$ and $\boldsymbol{\gamma}$ can be expressed as functions of covariates through, for instance, MLPs. Moreover, the normalising flows component can seamlessly integrate additional covariates by conditioning on them, as demonstrated by architectures like Masked Autoregressive Flows (Papamakarios et al., 2017).

Despite these strengths, GPDFlow has three practical limitations that we address here. Firstly, while GPDFlow benefits from the theoretically robust tail properties of the mGPD framework, it is constrained by the max-stable assumption in (4.1). Under this assumption, GPDFlow always exhibits asymptotic dependence regardless of the structure of the normalising flows, as evidenced by the positive upper tail coefficient $\chi_{1:d}$ in Propositions 9 and 10. Although GPDFlow can behave similarly to asymptotic independence (e.g., in the bivariate case, for sufficiently small values of $\mathbb{E}(\exp\{-|T_1 - T_2|\})$ so that the bivariate tail coefficient approaches zero), the steep decline in $\chi_{1:d}(q)$ as $q \to 1^-$ for asymptotically independent data (Huser and Wadsworth, 2019) typically requires setting a very high threshold (e.g., the 0.99-quantile or higher) in GPDFlow for a good approximation. Consequently, estimation accuracy can decrease due to increased variance and instability caused by limited exceedance data. This limitation particularly affects high-dimensional scenarios such as spatial extremes, where asymptotic independence often occurs, especially at distant locations. A practical solution is to apply GPDFlow exclusively to datasets known to exhibit asymptotic dependence (in the spatial context and depending on the data, this could be data from adjacent or close regions) and check if exceedance data are well-approximated by GPDFlow. A quick way to verify the suitability of GPDFlow is to use Algorithm (6) to identify the existence of stable trends in $\chi_{1:d}(q)$ and $\omega_{1:d}(q)$ at high thresholds.

Secondly, estimating uncertainty for marginal parameters in GPDFlow is challenging, despite these parameters being maximum likelihood estimates. Computing statistical uncertainty typically requires the inverse of the Hessian matrix of all parameters, which becomes computationally impractical given the large parameter space (often thousands) involved in the normalising flows. A bootstrap approach may provide uncertainty estimates, but at a substantial computational cost.

Lastly, accurate and stable GPDFlow estimates require threshold exceedance data to be on a similar scale. Divergent data scales can hinder gradient descent convergence or produce inaccurate estimates, a common issue in deep learning models (Ioffe and Szegedy, 2015). To mitigate this, data rescaling can be performed prior to model training. Specifically, fitting a univariate GPD to each component's exceedance data and transforming the data using the estimated parameters $\widehat{\sigma}_i$ and $\widehat{\gamma}_i$ via (4.6) helps improve training stability. While this method compromises joint marginal and dependence modelling, it preserves the flexible dependence modelling capability of GPDFlow.

4.7 Supplementary material

4.7.1 Code and data

The code and data required to reproduce the results in Sections 4.4 and 4.5 are freely available at `https://github.com/hcl516926907/GPDFlow.git`.

4.7.2 Proofs

Proof of Proposition 9. The first part of Proposition 9 is a special case $d = 2$ of Proposition 10, so its proof is omitted here.

For the second part, let $X_1 = \exp\{T_1 - \max(\boldsymbol{T})\}$ and $X_2 = \exp\{T_2 - \max(\boldsymbol{T})\}$. By the exchangeability assumption of T_1 and T_2, we have $\mathbb{E}(X_1) = \mathbb{E}(X_2) := c$. Since X_1 and X_2 can only be pair $(1,\ \exp\{-|T_1 - T_2|\})$ or $(\exp\{-|T_1 - T_2|\},\ 1)$,

$$\mathbb{E}(X_1 + X_2) = 1 + \mathbb{E}(\exp\{-|T_1 - T_2|\}) = 2c$$

Hence $c = \frac{1+\mathbb{E}(\exp\{-|T_1-T_2|\})}{2}$. Now

$$\begin{aligned}
\chi_{1,2} &= \mathbb{E}\left(\min\left\{\frac{\exp\{T_1 - \max(\boldsymbol{T})\}}{\mathbb{E}(\exp\{T_1 - \max(\boldsymbol{T})\})}, \frac{\exp\{T_2 - \max(\boldsymbol{T})\}}{\mathbb{E}(\exp\{T_2 - \max(\boldsymbol{T})\})}\right\}\right)\\
&= \mathbb{E}\left(\min\left\{\frac{X_1}{\mathbb{E}(X_1)}, \frac{X_2}{\mathbb{E}(X_2)}\right\}\right)\\
&= \frac{1}{c}\mathbb{E}\left(\min\left\{1, \exp\{-|T_1 - T_2|\}\right\}\right)\\
&= \frac{\mathbb{E}(\exp\{-|T_1 - T_2|\})}{c}\\
&= \frac{2\mathbb{E}(\exp\{-|T_1 - T_2|\})}{1 + \mathbb{E}(\exp\{-|T_1 - T_2|\})}
\end{aligned}$$

□

Proof of Proposition 10. The tail copula (Schmidt and Stadtmüller, 2006) $R(\boldsymbol{x}) : [0,\infty)^d \to [0,\infty)$ defined as

$$R(\boldsymbol{x}) = \lim_{n\to\infty} n\mathbb{P}\{F_1(X_1) > 1 - x_1/n, \cdots, F_d(X_d) > 1 - x_d/n\}$$

is associated to the stdf $\ell(\boldsymbol{x})$ by inclusion-exclusion formula and can be expressed as

$$R(\boldsymbol{x}) = \mathbb{E}(\min(\boldsymbol{x}\boldsymbol{V}))$$

by minimum–maximum identity and the same random vector $\boldsymbol{V}$ in (4.3) (Rootzén et al., 2018b). By Proposition 7.1 in the same paper, when $q > q^*$, $F_j(X_j) > 0, j = 1, \cdots, d$

$$\begin{aligned}
\chi_{1:d}(q) &= \frac{\mathbb{P}\{\bigcap_{j=1}^{d}\{X_j > F_j^{-1}(q)\}\}}{1-q} \\
&= \frac{R(q, \cdots, q)}{q} \\
&= \mathbb{E}(\min(\boldsymbol{V})) \\
&= \mathbb{E}\left(\min\left\{\frac{\exp\{T_1 - \max(\boldsymbol{T})\}}{\mathbb{E}(\exp\{T_1 - \max(\boldsymbol{T})\})}, \cdots, \frac{\exp\{T_d - \max(\boldsymbol{T})\}}{\mathbb{E}(\exp\{T_d - \max(\boldsymbol{T})\})}\right\}\right).
\end{aligned}$$

Similarly,

$$\begin{aligned}
\omega_{1:d}(q) &= \frac{\mathbb{P}\{\bigcup_{j=1}^{d}\{X_j > F_j^{-1}(q)\}\}}{1-q} \\
&= \frac{\ell(q, \cdots, q)}{q} \\
&= \mathbb{E}(\max(\boldsymbol{V})) \\
&= \mathbb{E}\left(\max\left\{\frac{\exp\{T_1 - \max(\boldsymbol{T})\}}{\mathbb{E}(\exp\{T_1 - \max(\boldsymbol{T})\})}, \cdots, \frac{\exp\{T_d - \max(\boldsymbol{T})\}}{\mathbb{E}(\exp\{T_d - \max(\boldsymbol{T})\})}\right\}\right)
\end{aligned}$$

□

4.7.3 Hyperparameter tuning results

We report the hyperparameter tuning results for GPDFlow in Simulations 1 and 2, as well as in the data application. Across all transformation layers, we use the same MLP architecture for both the log-scale and translation functions. Each MLP consists of one input layer, one hidden layer, and one output layer, where the input and output layers have dimensions equal to the data dimension d, and the hidden layer dimension is selected from $(2d, 4d, 6d, 8d)$. The number of transformations is tuned over $(4, 8, 12, 16)$. The negative log-likelihood (NLL) is monitored on the validation set, and early stopping is triggered if no improvement is observed for 50 consecutive epochs.

All models are trained using minibatches of size 64. Model parameters are updated using the Adam optimizer, with learning rates of 10^{-2} for the MLP parameters and 10^{-1} for the log-scale and shape parameters arising from the mGPD. A cosine annealing schedule is applied to all learning rates. For numerical stability, the gradient norm of all parameters is clipped to a maximum value of 1.

Simulation 1

Table 4.1: Hyperparameter tuning results for GPDFlow in Simulation 1. A grid search is performed over the number of flow transformations and the hidden-layer size of the one-layer MLPs used to parameterise the scale and translation functions in the Real NVP. The average negative log-likelihood (NLL) on the validation set, evaluated at the optimal epoch (50 epochs prior to early stopping), is reported.

		d=2		d=3		d=5	
# of transformation	# of hidden neurons	NLL	Best epoch	NLL	Best epoch	NLL	Best epoch
4	2*d	2.3	120	3.49	83	7.76	26
4	4*d	2.29	87	3.42	95	7.98	19
4	6*d	2.27	103	3.48	147	7.84	16
4	8*d	2.29	91	3.48	100	7.99	23
8	2*d	2.27	41	3.49	94	7.75	11
8	4*d	2.31	31	3.38	159	7.84	31
8	6*d	2.29	140	3.45	89	7.94	21
8	8*d	**2.24**	**58**	**3.35**	**92**	8.02	19
12	2*d	2.28	75	3.45	71	7.87	9
12	4*d	2.28	105	3.41	73	7.91	61
12	6*d	2.26	49	3.38	59	7.89	16
12	8*d	2.25	71	3.41	124	**7.74**	**33**
16	2*d	2.27	97	3.43	88	7.9	9
16	4*d	2.25	61	3.38	75	7.97	31
16	6*d	2.25	124	3.41	56	7.84	16
16	8*d	2.27	58	3.46	33	8.14	33

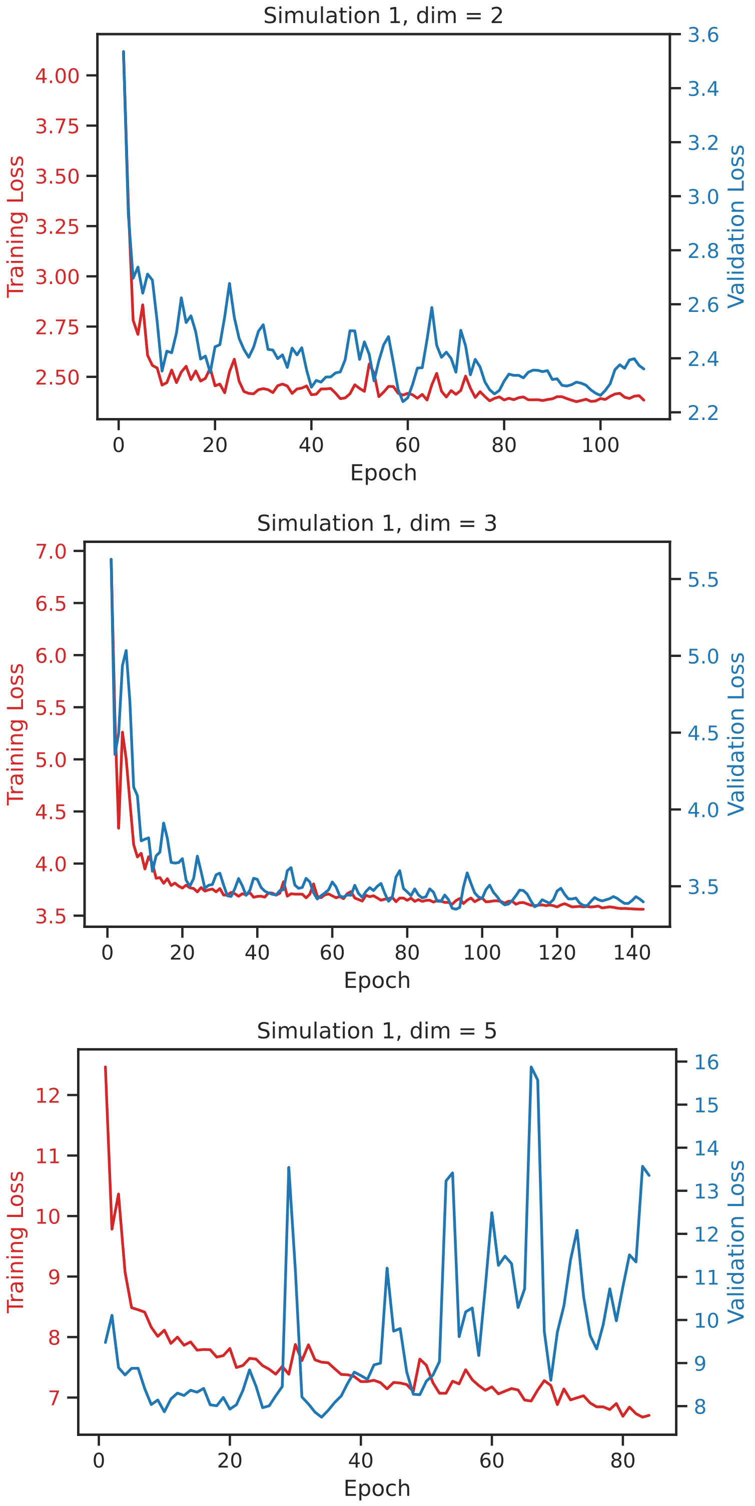

Figure 4.7: Learning curves for the models in Simulation 1, plotted over epochs prior to early stopping. The average negative log-likelihood is shown in red for the training set and in blue for the validation set.

Simulation 2

Table 4.2: Hyperparameter tuning results for GPDFlow in Simulation 2. A grid search is performed over the number of flow transformations and the hidden-layer size of the one-layer MLPs used to parameterise the scale and translation functions in the Real NVP. The average negative log-likelihood (NLL) on the validation set, evaluated at the optimal epoch (50 epochs prior to early stopping), is reported.

# of transformation	# of hidden neurons	NLL	Best epoch
4	4	6.46	119
4	8	6.5	120
4	12	6.48	133
4	16	6.5	110
8	4	6.52	89
8	8	6.48	129
8	12	6.46	138
8	16	6.49	103
12	4	6.53	141
12	8	6.52	137
12	12	6.5	125
12	16	**6.44**	**110**
16	4	6.51	156
16	8	6.55	141
16	12	6.48	111
16	16	6.49	163

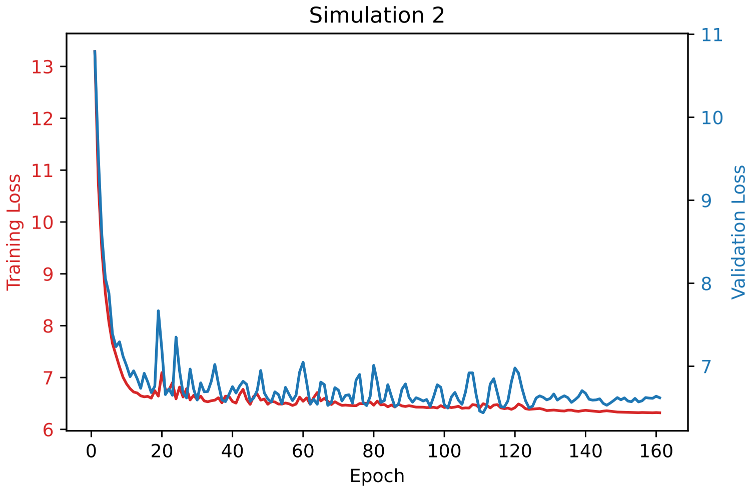

Figure 4.8: Learning curves for the models in Simulation 2 plotted over epochs prior to early stopping. The average negative log-likelihood is shown in red for the training set and in blue for the validation set.

Application

Table 4.3: Hyperparameter tuning results for GPDFlow in the data application. A grid search is performed over the number of flow transformations and the hidden-layer size of the one-layer MLPs used to parameterise the scale and translation functions in the Real NVP. The average negative log-likelihood (NLL) on the validation set, evaluated at the optimal epoch (50 epochs prior to early stopping), is reported.

# of transformation	# of hidden neurons	NLL	Best epoch
4	4	-8.29	70
4	8	**-8.60**	**109**
4	12	-8.13	55
4	16	-7.98	76
8	4	-8.23	66
8	8	-8.19	72
8	12	-7.41	42
8	16	-8.28	80
12	4	-7.82	71
12	8	-8.00	58
12	12	-8.24	54
12	16	-8.07	58
16	4	-7.61	74
16	8	-8.07	56
16	12	-8.06	52
16	16	-8.03	63

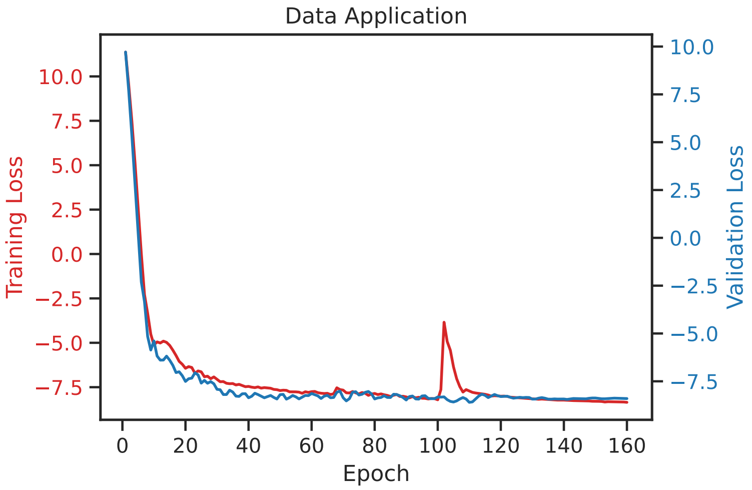

Figure 4.9: Learning curves for the models in the data application, plotted over epochs prior to early stopping. The average negative log-likelihood is shown in red for the training set and in blue for the validation set.

4.7.4 Supporting plots

Below are the plots that are not shown in the main body for space considerations.

Figure 4.5 continued

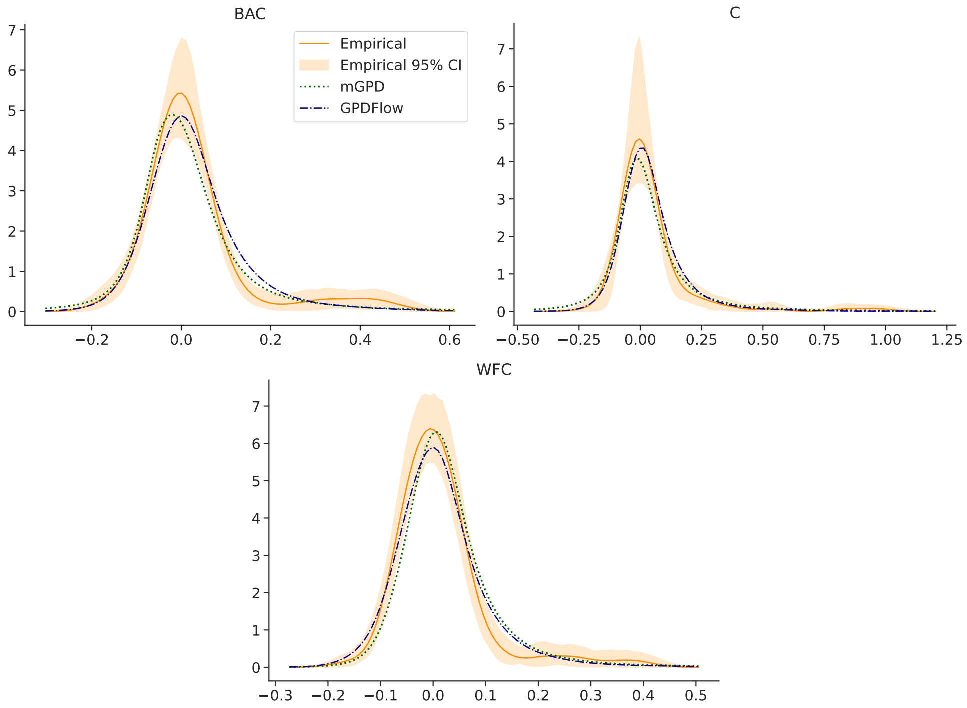

Figure 4.10: Marginal density estimation comparisons for bank BAC, C and WFC.

Chapter 5

Moderate and extreme wildfire forecast

Wildfires pose a major threat to Portugal, with over 115,000 hectares burned annually on average during 1980-2024, and the country has faced devastating mega-fires such as those in 2017. Accurate forecasts of wildfire occurrence and burned area are therefore essential for firefighting resource allocation and emergency preparedness. In this study, we propose a novel two-stage ensemble that extends the widely used latent Gaussian modelling framework with integrated nested Laplace approximation (INLA) for spatio-temporal wildfire forecasting. Stage 1 applies a gradient boosting model (XGBoost) to environmental covariates and historical fire records to produce one-month-ahead point forecasts of fire counts and burned area. Stage 2 uses these predictions as external covariates in a latent Gaussian model with additional spatiotemporal random effects to generate probabilistic forecasts of monthly total fire counts and burned area at the council level. To capture both moderate and extreme events, we implement the extended generalised Pareto (eGP) likelihood (a sub-asymptotic distribution) within INLA, develop Penalised Complexity (PC) priors for its parameters, and compare the eGP likelihood with common alternatives (e.g., Gamma and Weibull). Our framework tackles the unavailability of future environmental covariates at prediction time and performs strongly for one-month-ahead forecasts.

5.1 Introduction

Wildfires affect a vast portion of the vegetated surfaces of the Earth and may be defined as unplanned and uncontrolled fires that quickly spread over the terrain. Wildfires are favoured by prolonged drought, heat waves and by hot, dry and windy weather, and they may be triggered by natural phenomena, such as lightning strikes, or by human activities, including negligence, arson, and various forms of accidental ignition.

Portugal is among the countries most severely affected by wildfires, due to its mild and humid winters followed by hot and dry summers, strong winds, and extensive areas with forests and shrublands. According to official records, in 2017 alone, more than 21,000 fire ignitions were recorded, resulting in over 540,000 hectares burned (DaCamara, 2024). In addition to the high number of fire events and extent of burnt area, Portugal regularly experiences mega-fires, i.e., individual wildfires with extreme consequences. For instance, the deadly wildfires of 17 June 2017 claimed at least 66 lives and affected more than 220,000 hectares in 24 hours. These types of events result not only in economic losses and human casualties but also cause substantial environmental damage, including widespread deforestation. Consequently, the development of an effective wildfire forecasting system is crucial for enabling early warnings and better allocation of firefighting resources (DaCamara et al., 2018).

The increasing demand for accurate wildfire modelling has led to a wide range of studies employing statistical and machine learning methods. These approaches can be broadly categorised based on how they represent wildfires. One common approach models wildfire as an event occurring at an ignition point, optionally with an associated burnt area. This perspective motivates point process models for fire ignitions (Xu and Schoenberg, 2011; Gabriel et al., 2017; Opitz et al., 2020; Woolford et al., 2021) and marked point processes where the burnt area serves as the mark (Tonini et al., 2017; Xi et al., 2019; Koh et al., 2023; de Rivera et al., 2024; Duvsten Östin, Hanna and Gasslander, Tilda, 2025). Alternatively, ignition events and associated burnt areas can be aggregated over spatial partitions, with models targeting either the total burnt area or both fire count and burnt area per unit, leading to areal modelling approaches (Opitz, 2023; Cisneros et al., 2024; Lawler and Shaby, 2024).

To capture the risk of extremely large burnt areas, extreme value theory (EVT) is often employed to estimate high quantiles of the burnt area distribution. The Generalised Pareto Distribution (GPD) is a widely used EVT-based model for peak-over-threshold methods and has been applied to model exceedance probabilities (Pimont et al., 2021; Richards et al., 2023; Koh et al., 2023). When modelling the entire distribution of burnt areas, however, GPD-based methods require an auxiliary distribution for values below the threshold, which can lead to discontinuities in the likelihood. Recent advances have introduced sub-asymptotic distributions, which offer continuous density, flexible tail behaviour, and theoretical justification within EVT (Papastathopoulos and Tawn, 2013; Naveau et al., 2016). These distributions have seen successful applications in domains such as precipitation (Naveau et al., 2016), landslides (Yadav et al., 2021), and wildfires (Cisneros et al., 2024; Lawler and Shaby, 2024).

Bayesian hierarchical models with latent Gaussian fields are a popular framework for high-dimensional spatial and spatio-temporal modelling (Opitz, 2017). While inference is traditionally performed via Markov Chain Monte Carlo (MCMC), the integrated nested Laplace approximation (INLA) offers a faster and accurate alternative for posterior approximation, especially in space-time applications (Rue et al., 2009). Several studies have applied INLA-based frameworks to wildfire modelling (Gabriel et al., 2017; Opitz et al., 2020; Koh et al., 2023). Machine learning methods have also been explored, including tree-based models (Koh, 2023; Cisneros et al., 2023) and neural networks (Richards and Huser, 2022; Richards et al., 2023; Cisneros et al., 2024).

Despite the breadth of existing research, practical wildfire forecasting methods remain limited. The development and spread of wildfires are strongly influenced by environmental factors such as humidity, temperature, wind, and vegetation type. Incorporating such information is essential for improving predictive accuracy. However, most current spatio-temporal frameworks (e.g. Koh et al., 2023; Cisneros et al., 2024) are designed for retrospective analysis, using covariates observed at time $t + 1$ to predict wildfires also occurring at time $t + 1$. This setup limits their use in real-time forecasting, as it assumes access to future covariates that would not be available at the times for which predictions are to be made. Furthermore, within the popular latent Gaussian modelling framework, the additive structure and the practical constraints on the number of hyperparameters restrict the inclusion of multiple covariates in INLA-based models. As a result, studies often rely on a small number of representative variables, such as the Fire Weather Index (FWI), an index developed by the Canadian Forestry Service that has proven to be an especially suitable indicator of meteorological fire danger in Mediterranean ecosystems (DaCamara et al., 2014; Pinto et al., 2018; Nunes et al., 2023)

In this work, we aim to address the twin challenges of acquiring future covariates and the limited capacity of the INLA framework to accommodate numerous predictors. We propose a two-stage, interpretable modelling framework that relies entirely on readily available reanalysis data for probabilistic forecasting, leveraging both machine learning and INLA. In the first stage, we train a tree-based ensemble model, specifically, XGBoost (Chen and Guestrin, 2016), on a window-based dataset, incorporating environmental covariates and historical wildfire data up to time t, with the target being wildfire activity at time $t + 1$. This model learns patterns from historical data to produce a point forecast for the next time step. In the second stage, the XGBoost forecast is used as a synthetic future covariate, combined with spatial and temporal Gaussian effects in a latent Gaussian model estimated via INLA, yielding posterior predictive distributions.

The XGBoost model effectively encodes the information from all available covariates into a single, most informative predictor for Portuguese wildfires. This strategy reduces the reliance on future environmental covariates, circumvents the limitations of INLA in handling a large number of predictors, and enhances the INLA model's ability to capture complex interactions among covariates. Meanwhile, INLA provides a principled Bayesian framework for uncertainty quantification through the posterior predictive distribution. This hybrid modelling framework can be viewed as a stacking strategy (Wolpert, 1992), which has been shown to yield substantial improvements in predictive performance (Van der Laan et al., 2007). In contrast to other hybrid approaches for spatial and spatio-temporal modelling that require specialised mappings between spatial processes or covariance structures (Wikle and Zammit-Mangion, 2023), our approach minimises modifications to the underlying spatial structure, thereby preserving interpretability and facilitating its application to other spatial or spatio-temporal modelling tasks within the INLA framework. Our proposed method is also closely related to residual-learning approaches (MacBride et al., 2025), but does not rely on iterative updates between the first-stage and second-stage models. In our setting, such iterative training would be computationally expensive due to the inclusion of complex spatio-temporal random effects in the INLA model, but the potential gains from iterative residual learning are likely to be marginal.

Additionally, we contribute to the integration of the extended Generalised Pareto (eGP) likelihood (Naveau et al., 2016) within the INLA framework. The eGP distribution belongs to the family of sub-asymptotic models capable of jointly modelling the bulk and tail of the data. We use this distribution to model burnt area data, thereby capturing the moderate and extreme fires simultaneously in a continuous manner. As part of our implementation, we derive and incorporate penalised complexity (PC) priors with a closed-form expression for the two shape parameters of the eGP distribution.

The remainder of the paper is structured as follows. Section 5.2 introduces the Portuguese wildfire dataset and the environmental covariates used. Section 5.3 presents the two-stage modelling framework, including the choice of priors for the eGP parameters. Section 5.4 reports forecasting results and model interpretation. Section 5.5 discusses the use of the eGP likelihood and considerations for longer-horizon forecasting. Finally, Section 5.6 concludes the study.

5.2 Data preparation

5.2.1 Wildfire data scope

The wildfire dataset used in this study is sourced from the Portuguese Institute for Nature Conservation and Forests (ICNF, https://www.icnf.pt/florestas/gfr/gfrgestaoinformacao/estatisticas), a governmental agency responsible for forest and conservation policy. The dataset includes detailed information on wildfire events, such as ignition time (year, month, day, hour), fire duration, geographical coordinates (longitude and latitude), burnt area by land type (urban, bush, or agricultural), and additional attributes including cause and FWI.

This study focuses on wildfire records from 2011 to 2023, a 13-year period that captures both typical wildfire activity and extreme events such as the 2017 mega-fires, while keeping the temporal dimension manageable for computational modelling. To ensure that the analysis excludes intentional land management fires (e.g. crop residue burning), only fire events with a total burnt area exceeding 1 hectare and a duration longer than 3 hours are retained.

Administrative metadata, including council and district information, is mapped to each fire record. Fires are then aggregated to the council-month level to facilitate areal modelling. This choice is motivated by two factors: (1) The recorded ignition coordinates lack high spatial precision, and repeated wildfires are often observed at the same location, making a regional modelling unit such as the council more appropriate. (2) Council-level predictions are more interpretable and actionable for policymakers than point-level estimates. As such, modelling directly on aggregated data is preferred over integrating point process models post hoc. The temporal aggregation to a monthly resolution is a deliberate balance between computational feasibility and practical relevance. Modelling at higher temporal resolution (e.g. daily or hourly) over a 13-year span would be computationally intensive and likely require oversimplified spatio-temporal structures, potentially compromising model accuracy. Monthly aggregation allows for a more complex spatio-temporal modelling without excessive computational burden (Krainski et al., 2018).

After preprocessing, the dataset comprises 278 councils over 156 months, resulting in 43,368 council-month observations. For each observation, the total fire count and total burnt area are computed and serve as the primary response variables in the modelling framework. Observations with no recorded fire activity are assigned zeros for both responses. Unless otherwise specified, “fire count" and “burnt area" hereafter refer to these council-monthly totals. This aggregation

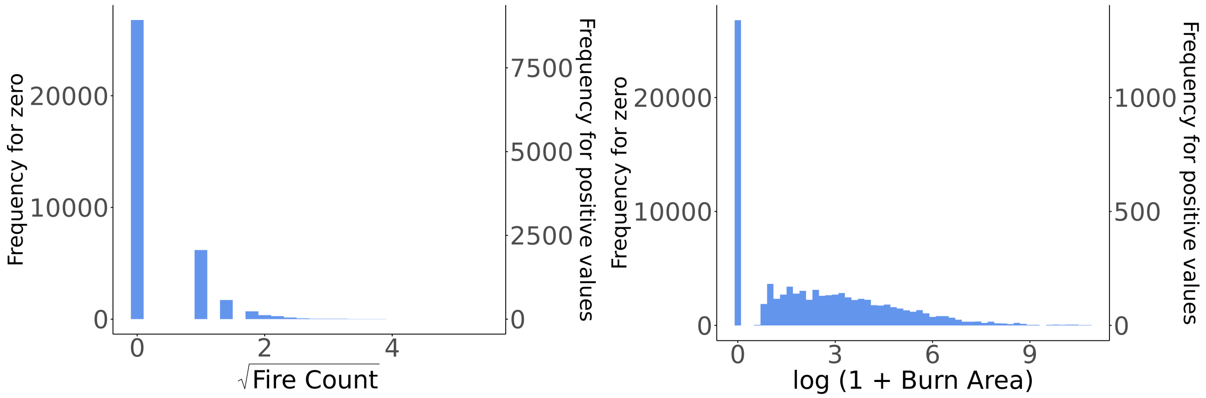

Figure 5.1: Histograms of fire count and burnt area at the council-month level, highlighting the prevalence of zeros. Quantities are rescaled for visualisation purposes using a square-root transformation for fire counts and a logarithmic transformation for burned area.

process naturally introduces a large number of zeros into the data, as illustrated in Figure 5.1, resulting in zero-inflation in both response variables. Additionally, both the fire count and burnt area distributions exhibit strong skewness, even after transformation (square root for fire count, logarithm for burnt area). Due to the pronounced skewness and heavy-tailed nature of the burnt area distribution, we model its square root to mitigate these effects. Alternative transformations for burnt area are discussed in Section 5.5.

Figure 5.2 presents the spatial and seasonal patterns of wildfires across Portugal. Spatial heterogeneity is evident: while fire count tends to increase with latitude and displays spatial autocorrelation among neighbouring councils, extreme burnt areas are more concentrated in central-north Portugal, with notable variability even between adjacent councils. Strong seasonal patterns are also observed, with peak activity occurring during summer and autumn.

5.2.2 Environmental covariates

Wildfire behaviour is strongly influenced by environmental conditions, particularly, climate related including, e.g., wind speed and direction, air temperature, and humidity. To enhance the predictive capabilities of our model, we incorporate 11 environmental covariates derived from reanalysis datasets spanning 2011–2023.

Five meteorological (air temperature, precipitation, wind speed and direction, dew point) and two vegetation-related covariates (green leaf area for two vegetation types) are obtained from ERA5-Land, which offers a fine spatial resolution ($0.1° \times 0.1°$). Four additional vegetation covariates, related to land cover types and their coverage percentages, are sourced from the ERA5, which has a relatively coarse resolution ($0.25° \times 0.25°$) and is time-invariant. In addition, two

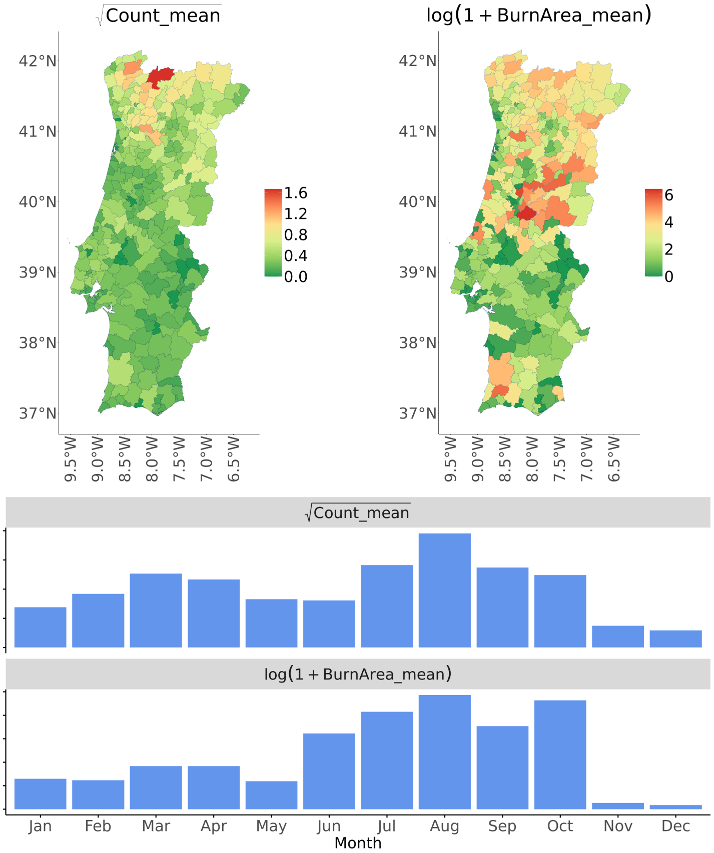

Figure 5.2: Top: Average council-level fire count and burnt area. Bottom: Monthly average fire count and burnt area across all councils. Spatial and seasonal variation is evident. Quantities are rescaled for visualisation purposes using a square-root transformation for fire counts and a logarithmic transformation for burned area.

derived covariates computed from the daily meteorological data are included: relative humidity and FWI.

All covariates are spatially mapped to the nearest council unit by assigning each grid cell to the council containing its centroid. Temporal aggregation is performed at the monthly level. For continuous variables, the aggregated mean is used; for categorical variables, the mode is applied. A complete list and description of all covariates are provided in Table 5.3 in the Supplementary Materials.

5.3 Methods

5.3.1 Overview

We propose a two-stage modelling framework to provide probabilistic forecasts of wildfires at the monthly council level. In the first stage, we employ the XGBoost algorithm to integrate historical wildfire activity data and current environmental covariates to generate one-month-ahead forecasts of wildfire count and burnt area for each council. These forecasts are then passed to a latent Gaussian model estimated via INLA, which incorporates spatio-temporal dependencies through council adjacency and temporal encoding. The INLA framework outputs full posterior predictive distributions for fire presence, fire count, and burnt area. To model the heavy-tailed behaviour of burnt areas, we implement the extended generalised Pareto (eGP) likelihood in INLA, which blends a Gamma-like left tail with a Pareto-like right tail, and is characterised by two shape parameters and a scale parameter (see Section 5.3.3 for details). This two-stage approach can be viewed as a form of model stacking (Wolpert, 1992), a type of ensemble learning where the predictions of one model are used as input features for another. Figure 5.3 provides a high-level overview of the proposed modelling pipeline.

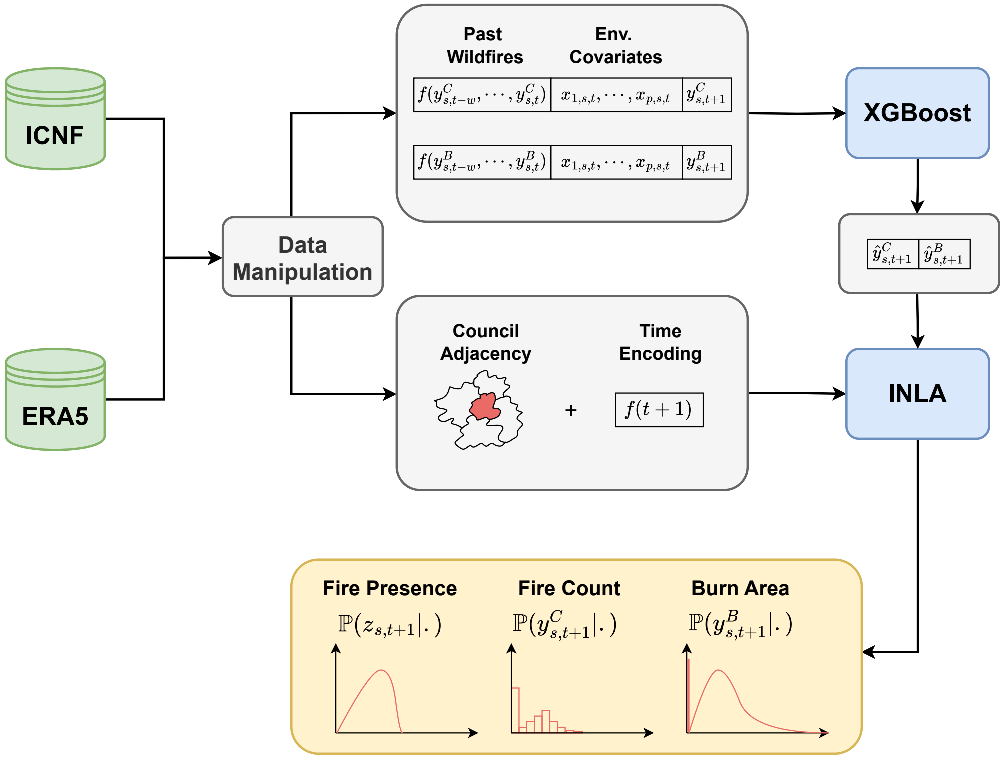
ICNF
ERA5
Data Manipulation
Past Wildfires
Env. Covariates
$f(y^C_{s,t-w},\cdots,y^C_{s,t})$
$x_{1,s,t},\cdots,x_{p,s,t}$
$y^C_{s,t+1}$
$f(y^B_{s,t-w},\cdots,y^B_{s,t})$
$x_{1,s,t},\cdots,x_{p,s,t}$
$y^B_{s,t+1}$
XGBoost
$\hat{y}^C_{s,t+1}$
$\hat{y}^B_{s,t+1}$
Council Adjacency
+
Time Encoding
$f(t+1)$
INLA
Fire Presence
$\mathbb{P}(z_{s,t+1}|.)$
Fire Count
$\mathbb{P}(y^C_{s,t+1}|.)$
Burn Area
$\mathbb{P}(y^B_{s,t+1}|.)$

Figure 5.3: Diagram of the proposed two-stage wildfire forecasting framework combining XGBoost and INLA.

The combination of XGBoost and INLA utilises the strengths of each framework while mitigating their individual limitations. XGBoost is particularly effective in modelling complex, nonlinear interactions among environmental and historical variables. By incorporating past wildfire activity in an autoregressive manner, the two XGBoost models, one for fire count and one for burnt area, can partially recover information that may be lost due to the spatial smoothing inherent in gridded environmental data. However, a key drawback of XGBoost is its lack of native uncertainty quantification. This is addressed in the second stage of our framework, where the INLA-based Bayesian hierarchical model provides full posterior distributions, allowing for straightforward uncertainty quantification.

5.3.2 Stage I: XGBoost

In stage I, we use two XGBoost models, both taking the accessible meteorological variables and past wildfire activities as features but one taking one-month ahead fire count as the target and the other one taking one-month ahead burnt area as the target, to generate two proxy covariates for the next level modelling. The choice of loss function ℓ is central to the performance of XGBoost and should align with the distributional properties of the response variable. In our case, fire count is a non-negative integer and is naturally modelled using a Poisson loss. By contrast, the burnt area is of a mixed type: it is continuous and positive when fires occur, but has a point mass at zero when no fire is observed. For this, we adopt the Tweedie deviance loss (Jørgensen, 1987), which corresponds to a compound Poisson-Gamma distribution. This distribution models a sum of Gamma-distributed fire sizes conditional on a Poisson-distributed number of occurrences:

$$Y = \begin{cases} 0 & \text{if } N = 0, \\ \sum_{i=1}^{N} X_i & \text{if } N = 1, 2, \dots \end{cases}$$

where N is a Poisson random variable, and X_i are i.i.d. Gamma random variables. Here, N and X_i can be interpreted as the council-month level fire count and the burnt area in each fire ignition, respectively, and Y represents the total burnt area at the council-month level. The corresponding Tweedie deviance loss function of true value y and prediction (mean of the Tweedie) $\widehat{y}$ is

$$\ell(y, \widehat{y}, k) = 2\left(\frac{\max\{y, 0\}^{2-k}}{(1-k)(2-k)} - \frac{y\widehat{y}^{1-k}}{1-k} + \frac{\widehat{y}^{2-k}}{2-k}\right), \quad 1 < k < 2,$$

where the index k governs the shape of the distribution: values closer to 1 approximate a Poisson, and those nearer 2 approach a Gamma.

Window-based modelling

Tree-based models, including XGBoost, do not inherently account for sequential dependencies in time-series data. A naive implementation would treat each time point independently, using covariates $\boldsymbol{x}_{t+1}$ to predict the target y_{t+1}, thereby neglecting temporal autocorrelation and requiring future covariates for forecasting. Given our 1-month modelling granularity, it is challenging to obtain the future environmental covariates over such a large horizon.

To address this limitation, we adopt a window-based approach, a common practice in time-series forecasting for non-sequential models (Elsayed et al., 2021). For a given forecasting horizon of one month, we reformulate the data into a lagged autoregressive structure: each target y_{t+1} is modelled as a function of covariates and wildfire records from previous time steps, such as $(\boldsymbol{x}_t, y_t, y_{t-1}, \ldots, y_{t-w+1})$, where w is the time window size. This configuration enables the model to learn temporal dependencies and make forecasts based on available historical wildfire data and covariates. It also helps mitigate the issue of smoothed covariate values, which are uninformative for local fire prediction, by incorporating recent wildfire history. Figure 5.4 contrasts the window-based modelling with the naive approach.

Autocorrelation plots (ACF) of the monthly fire count and burnt area in Figure 5.13 in the Section 5.7 reveal short-term dependencies and seasonal peaks at lags 12, 24, and 36, suggesting strong annual cycles. Based on this, we set $w = 36$, and include both short and long-term lag features. For short-term features, past fire count and burnt area up to lag 9 are included. Long-term and periodic patterns are captured by feature-engineered covariates. A full list of autoregressive features is provided in Table 5.4 in Section 5.7.

Let $\widetilde{\boldsymbol{x}}_{s,t}^{C}$ and $\widetilde{\boldsymbol{x}}_{s,t}^{B}$ denote the complete feature vectors used to forecast fire count (C) and burnt area (B), respectively, at council s and time t. Both vectors share the same covariates listed in Table 5.3 and 5.4, except $\widetilde{\boldsymbol{x}}_{s,t}^{C}$ includes only fire count-based autoregressive covariates, while $\widetilde{\boldsymbol{x}}_{s,t}^{B}$ includes only those based on burnt area. The forecasts for fire count and burnt area at council s and time $t+1$ are then given by:

$$\widehat{y}_{s,t+1}^{C} = \sum_{m_1} f_{m_1}^{C}(\widetilde{\boldsymbol{x}}_{s,t}^{C}), \tag{5.1}$$

$$\sqrt{\widehat{y}_{s,t+1}^{B}} = \sum_{m_2} f_{m_2}^{B}(\widetilde{\boldsymbol{x}}_{s,t}^{B}), \tag{5.2}$$

where $f_{m_1}^{C}$ and $f_{m_2}^{B}$ are regression trees trained to minimise Poisson and Tweedie deviance losses, respectively.

Output generation

We adopt a time-adjusted Super Learner cross-validation (CV) scheme (Van der Laan et al.,

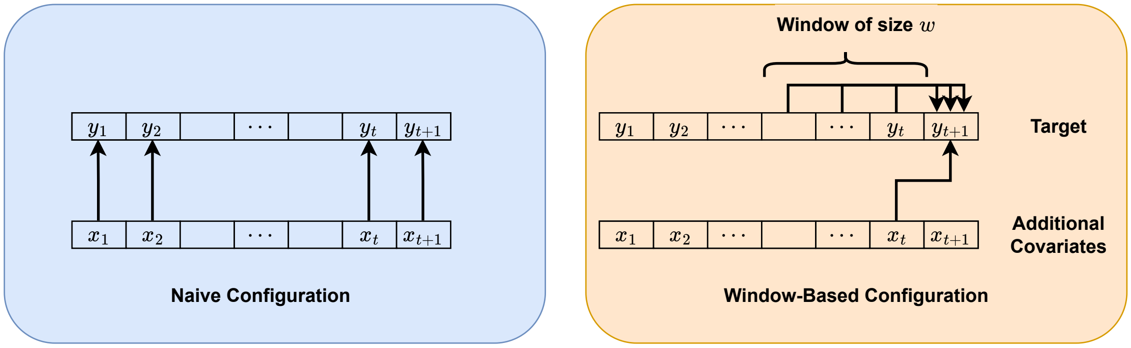

Figure 5.4: Comparison of naive and window-based modelling configurations. Arrows indicate the direction of information flow from predictors to targets.

2007) to tune hyperparameters and generate one-month-ahead forecasts of fire counts and burnt area. Standard k-fold CV is unsuitable for time series, as random splits may allow training on observations that occur after the validation set, thereby introducing look-ahead bias. To avoid this, we employ a chronological CV strategy, in which the model is trained exclusively on past folds and evaluated on subsequent future folds.

Formally, let $D_k, k = 1, \ldots, 12$ denote the subsets of records corresponding to calendar years 2011–2022 in the training set, and let k_0 mark the end of the warm-up period required to accumulate sufficient history for stable forecasting in the first validation fold. For each validation year $k_{\text{val}} > k_0$, we fit the model on $\bigcup_{k<k_{\text{val}}} D_k$ and evaluate it on $D_{k_{\text{val}}}$. Hyperparameters are selected by optimising metrics computed on pooled out-of-fold predictions, i.e. aggregating predictions from all validation years $k_{\text{val}} > k_0$ and evaluating the metric once on this combined set. After selection, we (i) refit on $\bigcup_{k<k_{\text{val}}} D_k$ to generate within-training one-month-ahead forecasts for $D_{k_{\text{val}}}$, and (ii) refit on $\bigcup_{k=1}^{12} D_k$ to forecast the test period (2023). This procedure prevents information leakage and provides performance estimates that reflect out-of-time generalisation, as illustrated in Figure 5.5.

5.3.3 Stage II: Bayesian latent Gaussian modelling

We incorporate the one-month-ahead forecasts of fire count and burnt area from equations (5.1) and (5.2) as external covariates in a Bayesian latent Gaussian model, which yields the final probabilistic predictions. This two-stage setup reflects the principles of stacked generalisation, where the predictions from one model serve as informative inputs to a second, more interpretable model to enhance overall performance.

The XGBoost forecasts compensate for the structural inflexibility of latent Gaussian models

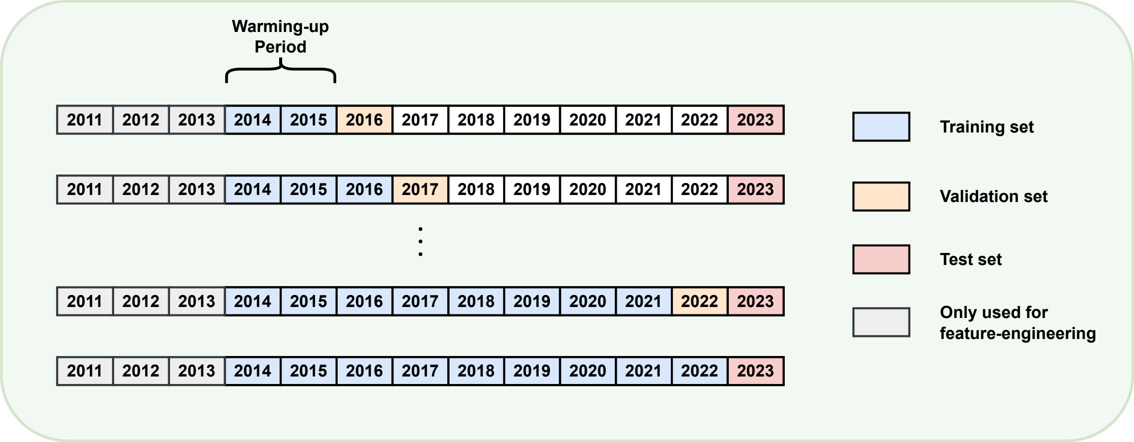

Figure 5.5: Time-series–aware cross-validation is implemented using yearly blocks for both hyperparameter tuning and the generation of one-month-ahead forecasts. The first three years are excluded from the training set because feature construction based on a 36-month rolling window results in missing values during this period.

in handling many covariates, particularly when those covariates have nonlinear interactions. Conversely, the Bayesian latent Gaussian model addresses a key limitation of XGBoost: the lack of native uncertainty quantification. The Bayesian latent Gaussian model provides full posterior distributions for quantities of interest, thus combining flexibility with interpretability and probabilistic reasoning.

In a Bayesian latent Gaussian model, observations $\boldsymbol{y} = (y_1, \ldots, y_N)$ are modelled in a hierarchical manner: $\boldsymbol{y}$ is assumed to be conditional independent given the linear predictor $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_N)$ and some hyperparameters $\boldsymbol{\theta}_1$ in the observational likelihood. The linear predictor $\boldsymbol{\eta}$ is further assumed to be a latent Gaussian field, whose mean vector and precision matrix are parameterised by parameters $\boldsymbol{\theta}_2$ with assigned priors. A general form of this structure is

$$\begin{aligned}
\boldsymbol{\theta}_2 &\sim \pi(\boldsymbol{\theta_2}) \\
\boldsymbol{\eta} \mid \boldsymbol{\theta}_2 &\sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\theta_2}), \mathcal{Q}(\boldsymbol{\theta}_2)^{-1}) \\
\boldsymbol{y} \mid \boldsymbol{\eta}, \boldsymbol{\theta_1}, \boldsymbol{\theta_2} &\sim \prod_{i=1}^{N} \pi(y_i \mid \boldsymbol{\eta}_i, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2).
\end{aligned}$$

Likelihood specification

Although Poisson and Tweedie likelihoods perform well in modelling general wildfire trends via XGBoost, they are not optimal in the latent Gaussian framework. Specifically, the Tweedie distribution, being composed of i.i.d. Gamma components, features a light right tail and is therefore unsuitable for capturing extreme burnt areas. Instead, we adopt the extended generalised Pareto (eGP) distribution (Naveau et al., 2016) for modelling the burnt area. This distribution combines features of Gamma and Pareto distributions by applying a power transformation to the standard Pareto distribution. It is defined on the positive real line and behaves like a Gamma for small values and like a Pareto for large values. The right tail behaviour is controlled by a shape parameter ξ, allowing flexibility in modelling heavy tails.

To handle zero-inflated data, we use a hurdle model, which separately models zero and positive outcomes. The hurdle model is defined as:

$$\pi(y) = \begin{cases} \mathbb{P}(Z = 0), & y = 0, \\ \mathbb{P}(Z = 1)\pi(y \mid Z = 1), & y > 0, \end{cases} \tag{5.3}$$

where Z is a latent Bernoulli variable indicating the presence of a non-zero event. The structure in (5.3) can be easily implemented in the latent Gaussian model framework for wildfire modelling by introducing an auxiliary Bernoulli variable Z of the same length as the observations. Conditional on $Z = 1$, the response y is modelled using a suitable distribution $\pi(y \mid Z = 1)$ with positive support.

For the fire count, the conditional distribution $\pi(y|Z = 1)$ is modelled using a zero-truncated Poisson distribution with parameter λ:

$$\mathbb{P}(Y = y; \lambda) = \frac{1}{1 - \exp(-\lambda)} \frac{\lambda^y}{y!} \exp(-\lambda), \quad y = 1, 2, \ldots$$

For the burnt area, we model the square root of the area using the eGP distribution. The cumulative distribution function of eGP is given by:

$$F(y; \sigma, \xi, \kappa) = \begin{cases} \left[1 - (1 + \xi y/\sigma)^{-1/\xi}\right]^{\kappa}, & y > 0,\ \xi \neq 0, \\ \left[1 - \exp(-y/\sigma)\right]^{\kappa}, & y > 0,\ \xi = 0, \end{cases} \tag{5.4}$$

where $\xi \in \mathbb{R}$ controls the rate of upper tail decay, $\sigma > 0$ is the scale, and $\kappa > 0$ governs the shape of the lower tail. The linear predictor η is linked to the α-quantile q_α of eGP by $q_\alpha = \exp(\eta)$, where α is typically set to be 0.5. By setting κ and ξ as hyperparameters, the scale parameter σ

can be expressed as a function of η, α, κ and ξ:

$$\sigma(\eta) = \frac{\xi q_\alpha}{(1-\alpha^{1/\kappa})^{-\xi}-1} = \frac{\xi \exp(\eta)}{(1-\alpha^{1/\kappa})^{-\xi}-1}.$$

We define three linear predictors for each council s at time $t+1$: $\eta^Z_{s,t+1}$, $\eta^C_{s,t+1}$ and $\eta^B_{s,t+1}$, corresponding to fire presence (Z), fire count (C) and burnt area (B), respectively. The full hierarchical model is described as:

$$\begin{aligned} Z_{s,t+1} \mid \eta^Z_{s,t+1} &\sim \text{Bernoulli}\{\text{logit}^{-1}(\eta^Z_{s,t+1})\} \\ \{Y^C_{s,t+1} \mid \eta^C_{s,t+1}, Z_{s,t+1}=1\} &\sim \text{Trucated Poisson}\{\exp(\eta^C_{s,t+1})\} \\ \left\{\sqrt{Y^B_{s,t+1}} \mid \eta^B_{s,t+1}, Z_{s,t+1}=1\right\} &\sim \text{eGP}\{\sigma\left(\eta^B_{s,t+1}\right), \xi, \kappa\} \\ \xi, \kappa &\sim \text{Hyperpriors.} \end{aligned}$$

Effects in the linear predictor

The latent effects in the linear predictors comprise Gaussian random effects derived from XGBoost predictions, as well as spatio-temporal Gaussian effects informed by adjacency structures and time.

Although the XGBoost predictions $\widehat{y}^C_{s,t+1}$ and $\widehat{y}^B_{s,t+1}$ could be included as fixed effects, doing so would impose a linear relationship with the linear predictor. Given the complex spatial and spatio-temporal dependencies present in the data, such a restriction is overly limiting. Instead, we treat the XGBoost outputs as smooth random effects, allowing the second-stage model to flexibly recalibrate the machine-learning predictions in a spatially coherent and uncertainty-aware manner. Specifically, we model the effect of $\widehat{y}^{(\cdot)}_{s,t+1}$ using a first-order random walk (RW1) prior on 20 discretised bins of the prediction $\widehat{y}^{(\cdot)}_{s,t+1}$:

$$R_k - R_{k-1} \sim \mathcal{N}(0, \tau_R^{-1}), \tag{5.6}$$

where R_k represents the effect of the k-th bin of the discretised covariate and τ_R is the precision parameter. As the fire count and burnt area contribute to the Poisson and eGP likelihoods only when fire is present (i.e., when $y^C_{s,t+1} > 0$) in the hurdle model training, we discretise $\widehat{y}^C_{s,t+1}$ and $\widehat{y}^B_{s,t+1}$ conditional on $y^C_{s,t+1} > 0$ when constructing the R_k in their respective linear predictors $\eta^C_{s,t+1}$ and $\eta^B_{s,t+1}$. By contrast, $\widehat{y}^C_{s,t+1}$ and $\widehat{y}^B_{s,t+1}$ are discretised unconditionally when constructing the corresponding R_k in $\eta^Z_{s,t+1}$.

We incorporate spatially structured and unstructured effects through the Besag–York–Mollié model, using the reparameterised BYM2 formulation (Riebler et al., 2016; Simpson et al., 2017), as implemented in R-INLA. A BYM2 effect $\boldsymbol{b}$ combines a scaled intrinsic conditional autoregressive (CAR) component $\boldsymbol{\delta}$ (with unit variance) and unstructured noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Here, $\boldsymbol{I}$ represents the identity matrix with dimension defined by the length of $\boldsymbol{\epsilon}$. The effect is defined as

$$\boldsymbol{b} = \frac{1}{\sqrt{\tau}}(\sqrt{1-\phi}\boldsymbol{\epsilon} + \sqrt{\phi}\boldsymbol{\delta}),$$

where τ is the precision parameter, and $\phi \in [0, 1]$ controls the balance between spatial structure ($\phi = 1$) and unstructured variation ($\phi = 0$). To reflect different spatial scales, we define two adjacency graphs based on Portugal's administrative boundaries: one at the council level (fine granularity) and the other at the district level (coarse granularity), resulting in two distinct BYM2 effects, termed $\boldsymbol{b}_c$ for the council-level effect and $\boldsymbol{b}_d$ for the district-level effect. If we further denote the covariance of $\boldsymbol{b}_c$ as Σ_c and the covariance of $\boldsymbol{b}_d$ as Σ_d, then

$$\boldsymbol{b}_c \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_c), \quad \boldsymbol{b}_d \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_d).$$

To introduce temporal dynamics, we group the spatial effects over time. This provides greater flexibility than additive spatial and temporal terms. We consider two grouping schemes: Group 1 is based on calendar month (i.e., periodic across years), capturing seasonality. On the other hand, Group 2 is based on unique time indices, intended to capture residual temporal patterns not explained by Group 1. For each group, we assume independent Gaussian priors:

$$\boldsymbol{t}_1 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_1), \quad \boldsymbol{t}_2 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_2),$$

where $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ represent the effects of unique indices in Group 1 and Group 2, respectively, and $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ are identity matrices with dimensions matching the length of $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$.

To manage model complexity, we group the council-level BYM2 effect $\boldsymbol{b}_c$ by Group 1 and the district-level BYM2 effect $\boldsymbol{b}_d$ by Group 2. This yields a council-level spatio-temporal effect G_c and a district-level spatio-temporal effect G_d:

$$G_c \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_c \otimes \boldsymbol{I}_1), \quad G_d \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_d \otimes \boldsymbol{I}_2),$$

with $\otimes$ denoting the Kronecker product.

We also include a pure temporal effect for the year T to capture annual variation, potentially driven by policy changes following major wildfire events. T is assigned a Gaussian prior with precision τ_T:

$$T \sim \mathcal{N}(0, \tau_T^{-1}).$$

Bringing all components together, the linear predictors for fire presence η^Z, fire count η^C and burnt area η^B are expressed as:

$$\begin{aligned}
\eta^Z_{s,t+1} &= \beta^Z_0 + \quad G_c(s, t+1; \tau_{G_c}, \phi_{G_c}) + \quad G_d(s, t+1; \tau_{G_d}, \phi_{G_d}) + T(t+1; \tau^Z_T) + R(\widehat{y}^C_{s,t+1}; \tau_R^{Z^C}) + R(\widehat{y}^B_{s,t+1}; \tau_R^{Z^B}),\\
\eta^C_{s,t+1} &= \beta^C_0 + \beta^C_1\, G_c(s, t+1; \tau_{G_c}, \phi_{G_c}) + \beta^C_2\, G_d(s, t+1; \tau_{G_d}, \phi_{G_d}) + T(t+1; \tau^C_T) + R(\widehat{y}^C_{s,t+1}; \tau^C_R),\\
\eta^B_{s,t+1} &= \beta^B_0 + \beta^B_1\, G_c(s, t+1; \tau_{G_c}, \phi_{G_c}) + \beta^B_2\, G_d(s, t+1; \tau_{G_d}, \phi_{G_d}) + T(t+1; \tau^B_T) + R(\widehat{y}^B_{s,t+1}; \tau^B_R).
\end{aligned} \tag{5.7}$$

Here, $\beta_0^{(\cdot)}$ are intercepts, and $\beta_1^{(\cdot)}, \beta_2^{(\cdot)}$ are scaling parameters controlling the contribution of the shared spatio-temporal effects to each predictor. For parameters and effects that involve subscripts and superscripts, subscripts indicate the associated effect, while superscripts specify the predictor. Note that we include both predicted fire count and burnt area in η^Z since both positive fire count and burnt area indicate a fire presence. In addition, the spatio-temporal effects G_c and G_d are shared across all three linear predictors. Sharing allows the spatio-temporal effects for fire count and burnt area to be informed by the full dataset rather than only the subset with fire occurrence. This reduces the risk of overparameterisation and mitigates uncertainty arising from the sparse nature of fire events.

Priors

We now elaborate on the priors used for the eGP likelihood hyperparameters (κ and ξ) as well as those associated with the linear predictor. For κ and ξ, we adopt Penalised Complexity (PC) priors following the framework of Simpson et al. (2017), which provide a principled mechanism to control model complexity by penalising deviations from a simpler base model. This deviation is measured via the Kullback–Leibler divergence (KLD; Kullback and Leibler 1951), and the corresponding distance is defined as $d = \sqrt{2\text{KLD}}$. An exponential prior is then placed on this distance, yielding a memoryless penalty on increasing model complexity. The PC prior for each parameter is obtained by transforming this exponential prior back to the original parameter scale.

In the special case of the standard GDP ($\kappa = 1$), Opitz et al. (2018b) derived a PC prior for $\xi > 0$, using the base model with $\xi = 0$. They computed the KLD between a GPD density $f_\xi(y) = f_{\text{GPD}}(y; \xi)$ and the base model $f_{\xi_0}(y) = f_{\text{GPD}}(y; \xi = 0)$ as:

$$\text{KLD}\{f_\xi \| f_{\xi_0}\} = \frac{\xi^2}{1-\xi}, \quad 0 \le \xi < 1. \tag{5.8}$$

Two options were then considered for deriving the PC prior for ξ: (1) using the exact expression in (5.8), or (2) using the approximation $\text{KLD}\{f_\xi \| f_{\xi_0}\} \approx \xi^2$ as $\xi \to 0$. These yield the following prior formulations:

$$\begin{aligned}
&\text{Option 1:} \quad \pi_1(\xi) = \lambda \exp\left\{-\frac{\lambda\xi}{(1-\xi)^{1/2}}\right\}\left\{\frac{1-\xi/2}{(1-\xi)^{3/2}}\right\}, && 0 \le \xi < 1,\\
&\text{Option 2:} \quad \pi_2(\xi) = \lambda \exp\{-\lambda\xi\}, && 0 \le \xi < 1,
\end{aligned}$$

where λ is the rate parameter of the exponential distribution, controlling the strength of penalisation. The two priors are nearly indistinguishable for large λ (e.g., $\lambda > 3$), but diverge notably for smaller values, as illustrated in Figure 3 of Opitz et al. (2018b).

For the eGP setting, we relax the constraint $\xi > 0$ to allow for negative values, which may be relevant in practice. The eGP density exhibits a Pareto-type tail whose heaviness is governed by the shape parameter ξ, independent of κ. For analytical tractability in deriving the KLD, we fix $\kappa = 1$, which leads to a form of KLD same as that in (5.8) except for the support. Given the limited prior knowledge on the sign of ξ, we adopt a symmetric PC prior centred at zero. Using a second-order expansion of (5.8) around $\xi = 0$, we obtain:

$$\pi(\xi) = \frac{\lambda \exp\{-\lambda|\xi|\}}{\int_{\xi_L}^{\xi_U} \lambda \exp\{-\lambda|x|\}\,\mathrm{d}x}, \quad \xi_L < \xi < \xi_U, \tag{5.9}$$

where ξ_L and ξ_U are lower and upper bounds for ξ selected to enforce desirable properties such as finite moments. The resulting prior is symmetric around $\xi = 0$, matches $\pi_2(\xi)$ on the positive half-line, and incorporates truncations to preserve key theoretical properties of the eGP. In the R-INLA implementation, we use $(\xi_L, \xi_U) = (-0.5, 0.5)$, ensuring a finite mean and variance and desirable asymptotic properties for the maximum likelihood estimator of ξ.

The natural base model for constructing a PC prior for the parameter κ in the eGP distribution is $\kappa = 1$, which corresponds to the standard GP distribution. The KLD between $f_\kappa(y) =$

$f_{\text{eGP}}(y;\kappa)$ and $f_{\kappa_1}(y) = f_{\text{eGP}}(y;\kappa=1)$ is given by

$$\text{KLD}\{f_\kappa \| f_{\kappa_1}\} = \log\kappa - \frac{\kappa-1}{\kappa}, \quad \kappa > 0, \tag{5.10}$$

with derivation provided in Section 5.7.2. Following the approach for deriving a PC prior for the parameter ξ, we may either use the exact KLD in (5.10), or apply a second-order Taylor expansion around $\kappa = 1$, which gives

$$\text{KLD}\{f_\kappa \| f_{\kappa_1}\} \approx \frac{1}{2}(\kappa-1)^2, \quad \kappa > 0. \tag{5.11}$$

Given a penalisation rate λ, the PC prior based on the exact KLD in (5.10) takes the form

$$\pi_1(\kappa) = \begin{cases} \frac{\lambda|\kappa-1|}{2\kappa^2\sqrt{2\log\kappa-2(\kappa-1)/\kappa}} \exp\left\{-\lambda\left(\sqrt{2\log\kappa - 2(\kappa-1)/\kappa}\right)\right\}, & \kappa > 0,\ \kappa \neq 1, \\ \lambda/2, & \kappa = 1, \end{cases} \tag{5.12}$$

whereas the PC prior based on the approximated KLD in (5.11) is given by

$$\pi_2(\kappa) \ = \frac{\lambda\exp(-\lambda|\kappa-1|)}{2-\exp(-\lambda)}, \qquad \kappa > 0.$$

Figure 5.6 displays $\pi_1(\kappa)$ and $\pi_2(\kappa)$ under various values of λ. When λ is large (e.g. $\lambda > 5$), both priors concentrates around $\kappa = 1$, exhibiting similar behaviour. As λ decreases, the mode of $\pi_1(\kappa)$ shifts leftwards towards zero and diverge as $\kappa \to 0$, whereas $\pi_2(\kappa)$ remains locally symmetric around $\kappa = 1$. This suggests that $\pi_1(\kappa)$ always shrinks κ towards a value in $(0, 1]$, and it is less suitable for expressing weakly informative priors over $\kappa > 1$ compared to $\pi_2(\kappa)$. To determine the more appropriate prior between $\pi_1(\kappa)$ and $\pi_2(\kappa)$, we consider the interpretation and functional role of κ. According to Naveau et al. (2016), κ governs the lower-tail behaviour of the cumulative distribution function of an eGP random variable Y via

$$\mathbb{P}(Y < y) \approx \text{constant} \times y^\kappa, \qquad \text{as } y \to 0^+.$$

Consequently, the corresponding density function $f_{\text{eGP}}(y)$ satisfies

$$f_{\text{eGP}}(y) \propto \kappa y^{\kappa-1}, \qquad \text{as } y \to 0^+.$$

This characterisation implies that κ plays a role analogous to the shape parameter in the Gamma or Beta distribution. Specifically, when $\kappa > 1$, f_{eGP} increases from zero to a mode, while for $0 < \kappa < 1$, the density exhibits a singularity at zero, sharply peaking near the origin. In environmental applications, data may be zero-inflated; however, the positive component is more frequently well modelled by an eGP distribution with $\kappa > 1$ than with $0 < \kappa < 1$, as illustrated in Figure 5.1. From this perspective, we seek a prior $\pi(\kappa)$ that does not favour the region $0 < \kappa < 1$, irrespective of λ. Accordingly, we adopt $\pi(\kappa) = \pi_2(\kappa)$ for fitting the eGP model.

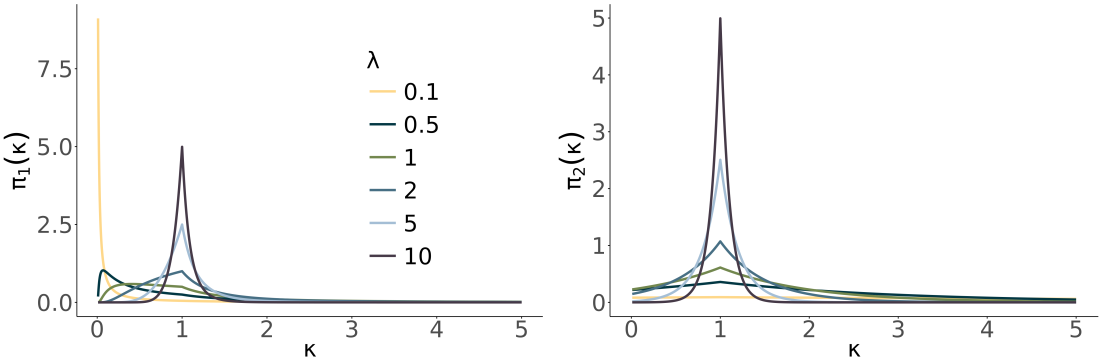

Figure 5.6: PC priors for κ based on exact KLD in (5.10) (left) and the approximated KLD around $\kappa = 1$ (5.11) (right) under penalisation rates $\lambda \in \{0.1, 0.5, 1, 2, 5, 10\}$.

The priors for ξ, κ, and the remaining hyperparameters in the linear predictor in (5.7) are specified as follows:

1. ξ and κ have priors in (5.9) and (5.12), respectively. In both cases, the penalisation rate parameter takes $\lambda = 10$.

2. Intercepts $\beta_0^{(\cdot)}$ are assigned weakly informative priors $\mathcal{N}(0, 1000)$;

3. Scaling parameters $\beta_1^{(\cdot)}$ are given more informative priors $\mathcal{N}(0, 0.1)$;

4. Precision parameters in $T(\cdot)$ and $R(\cdot)$ follow priors Gamma$(0.1, 0.1)$;

5. PC priors are used for parameters in $G_c(\cdot)$ and $G_d(\cdot)$, with constraints $\mathbb{P}(1/\sqrt{\tau} > 1) = 0.01$ and $\mathbb{P}(\phi < 0.5) = 0.5$ to control the marginal standard deviation and spatial range, respectively.

Here, we adopt Gamma priors rather than PC priors for the parameters in $T(\cdot)$ and $R(\cdot)$ because numerical instabilities were observed during model fitting across a wide range of PC prior specifications, from weakly informative to more informative choices. These instabilities were substantially mitigated after switching to Gamma priors.

5.4 Results

5.4.1 Model comparison

To evaluate the effectiveness of incorporating XGBoost-based wildfire forecasts into our modelling framework and to assess the appropriateness of the eGP likelihood for wildfire modelling, we compare several variations of the latent Gaussian model differing in their likelihood choices and linear predictor components. Let M1 denote the full two-stage model introduced in Section 5.3. To examine the role of the eGP likelihood, we substitute it with two alternative distributions commonly used in environmental applications: the Gamma and Weibull likelihoods, yielding models M2 and M3, respectively. Additionally, we construct M4 to emulate a setting in which future environmental covariates are unavailable. Specifically, the XGBoost-derived random effects are replaced by lag-1 values of the FWI and temperature. We retain FWI because it is one of the most widely used fire danger indicators, and temperature due to its high feature importance, as shown in Figure 5.8. All remaining random effects are kept identical to those in M1.

We use wildfire data from 2011 to 2022 as the training set and data from 2023 as the test set. One-month-ahead forecasts of fire count and burnt area are generated as described in Section 5.3.2. The posterior predictive distributions of the fire presence $Z_{s,t+1}$, fire count $Y^C_{s,t+1}$ and square-root-transformed burnt area $\sqrt{Y^B_{s,t+1}}$ are obtained from 1000 posterior simulations. To evaluate performance, we use Area Under Curve (AUC) to assess the predictive accuracy for fire presence, where the posterior mean of $\widehat{Z}_{s,t+1}$ serves as the estimate of $\mathbb{P}(Z_{s,t+1} = 1)$. The positive burnt area $\sqrt{\widehat{Y}^B_{s,t+1}} \mid Z_{s,t+1} = 1$ is evaluated by continuous ranked probability score (CRPS), which is a proper scoring rule that measures the difference between a predictive distribution F and a single observation y by

$$\text{CRPS}(F, y) = \int_{\mathbb{R}} \left[F(v) - \mathbb{1}(v \geq y)\right]^2 \mathrm{d}v,$$

where $\mathbb{1}$ is the indicator function. CRPS is computed for each location–time pair and averaged over all instances.

Since forecasts are typically communicated as probabilities over categorised bins, with particular emphasis on large events, we also compute weighted binned scores following the structure proposed by Opitz (2023). Specifically, for a fire count threshold vector $\boldsymbol{h}^C = (0, 1, 2, \ldots, 10, 15, 20, 25, 30)$, the weighted scoring for fire count r^C is the weighted sum of squared residuals between predicted and empirical probabilities across all observations

Table 5.1: Comparison of posterior predictive performance across model variants on the test set. AUC evaluates fire presence predictions (higher is better). Lower values indicate better performance for CRPS and binned scores. All results are based on 1,000 posterior predictive samples. Bold values highlight relatively better performance.

Metric	M1 (eGP)	M2 (Gamma)	M3 (Weibull)	M4 (eGP, lagged covariates)
AUC	**0.862**	0.857	0.860	0.815
CRPS	**4.75**	4.83	4.78	4.89
Unweighted r^C	**362**	371	366	382
Weighted r^C	**4.84**	4.93	4.88	5.14
Unweighted r^B	**565**	578	577	594
Weighted r^B	**14.2**	14.5	14.6	14.7

and thresholds:

$$r^C = \sum_{s,t+1} \sum_{h \in \boldsymbol{h}^C} w^C(h) \left[\widehat{\mathbb{P}}(Y^C_{s,t+1} \leq h) - \mathbb{1}(Y^C_{s,t+1} \leq h) \right]^2,$$

where the normalised weight is given by $w^C(h) = \widetilde{w}^C(h)/\widetilde{w}^C(30)$, and the unnormalised weights are defined as

$$\widetilde{w}^C(h) = 1 - (1 + (h+1)^2/1000)^{-1/4},$$

which increases approximately linearly with h. Here, $\mathbb{P}(Y^C_{s,t+1} \leq h)$ is the unconditional probability of $Y^C_{s,t+1}$ obtained by integrating out $Z_{s,t+1}$ from the conditional distribution $Y^C_{s,t+1} \mid Z_{s,t+1} = 1$.

Similarly, for burnt area (on the original scale), the thresholds are defined as $\boldsymbol{h}^B = (0, 20, 40, 60, 80, 100, 200, 300, 400, 500, 1000, 2000, 5000, 10000, 20000, 50000)$. The corresponding weighted binned score r^B is given by

$$r^B = \sum_{s,t+1} \sum_{h \in \boldsymbol{h}^B} w^B(h) \left[\widehat{\mathbb{P}}(Y^B_{s,t+1} \leq h) - \mathbb{1}(Y^B_{s,t+1} \leq h) \right]^2,$$

where the normalised weights are defined as $w^B(h) = \widetilde{w}^B(h)/\widetilde{w}^B(50000)$, $\widetilde{w}^B(h) = 1 - (1 + (h+1)/1000)^{-1/4}$. Similar to $w^C(h)$, $w^B(h)$ increases approximately linearly with the threshold h, placing more emphasis on larger fire events.

Table 5.1 summarises the model performance on the test set in 2023 across six evaluation metrics. Among models M1, M2, and M3, which differ only in the likelihood, the eGP likelihood achieves slightly better performance across all metrics. M1 and M3 perform comparably, as both eGP and Weibull likelihoods can accommodate heavy-tailed data through their shape parameter.

In contrast, M2 employs a light-tailed distribution (Gamma) for burnt area and performs worst on five of the six metrics. Notice that although the likelihood is modified only for burnt area, changes also propagate to metrics such as AUC and binned scores for counts, owing to the shared spatio-temporal effects in the joint modelling of fire presence, fire count, and burnt area. Nonetheless, performance differences between M1, M2, and M3 are marginal, and the reasons for this will be further discussed in Section 5.5.2. For consistency, we proceed with eGP (M1) in the remainder of the paper.

Model M4, which excludes XGBoost-derived covariates, relies only on lagged temperature and FWI covariates together with spatio-temporal effects in the latent Gaussian model to produce one-month-ahead forecasts. A substantial deterioration in performance is observed across all six evaluation metrics, with particularly pronounced declines in metrics that weight all events equally, such as AUC, CRPS, and the unweighted correlations r^C and r^B. These results underscore the importance of incorporating dynamic, forecast-driven covariates that capture complex spatio-temporal structure for accurate wildfire forecasting.

5.4.2 Posterior predictions

Figure 5.7 presents a detailed view of the posterior predictive distributions, $\pi(\widehat{Y}^C_{s,t+1})$ and $\pi(\widehat{Y}^B_{s,t+1})$, obtained from the second stage model. To assess predictive performance, we conduct posterior predictive checks based on the threshold exceedance probabilities of fire count and burnt area within the test set. The empirical exceedance probabilities at a given threshold h are defined as

$$\widehat{P}(Y^C > h) = \frac{1}{|\mathcal{I}|} \sum_{(s,t+1)\in\mathcal{I}} \mathbb{1}(Y^C_{s,t+1} > h), \quad \widehat{P}(Y^B > h) = \frac{1}{|\mathcal{I}|} \sum_{(s,t+1)\in\mathcal{I}} \mathbb{1}(Y^B_{s,t+1} > h),$$

where $\mathcal{I}$ denotes the set of all spatial and temporal indices in the test set, and $|\mathcal{I}|$ is its cardinality. These quantities represent the overall proportions of exceedances in the test set, aggregated over space and time, and are not conditioned on specific locations or time points.

The upper panels of Figure 5.7 show the posterior predictive check of the exceedance probabilities over various thresholds based on 1000 posterior predictive replicates. Uncertainty is notably higher at lower thresholds (e.g., 5 fires or 10 hectares) and diminishes as the threshold increases. When compared with empirical exceedance probabilities, the model performs well for burnt area: the observed values (red dashed lines) consistently fall within the 50th–90th percentiles of the posterior predictive distribution. For fire counts, some deviations occur at lower thresholds, but the observed statistics do not systematically fall outside the predictive envelopes.

The lower panels of Figure 5.7 display the posterior predictive distributions for the total fire count and burnt area across Portugal at selected time points. We focus on one year from the training set (2017), during which several severe wildfires occurred, and a full year from the test set (2023), to evaluate the model's ability to capture temporal dynamics. Each box plot is generated from 1000 posterior samples, with aggregated totals over the entire Portugal and specified time periods.

Overall, the results demonstrate that the model effectively captures the temporal evolution trend of wildfire activity. For most time points, the observed fire counts and burnt areas lie within 1.5 times the interquartile range (IQR) from the first and third quartiles of the posterior predictive distributions. Notably, the model yields accurate predictions for October 2017, when Portugal experienced an exceptionally intense wildfire episode, with over 350 reported fires and a burnt area exceeding 250,000 hectares.

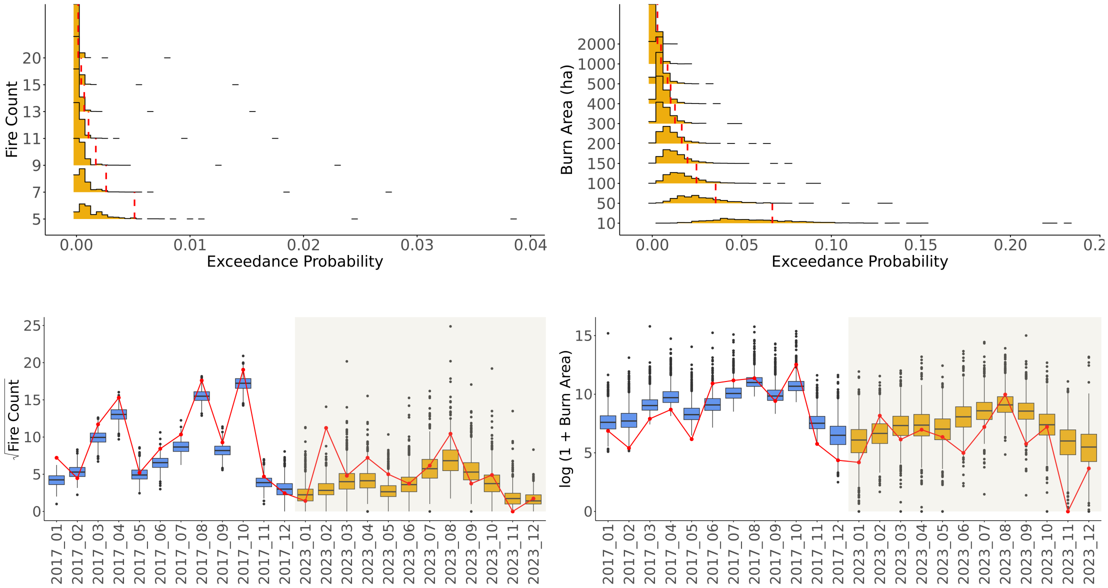

Figure 5.7: Posterior predictive checks of threshold exceedance probabilities for fire count and burnt area in the test set (top two panels), and posterior predictive distributions of total fire count and burnt area in Portugal for 2017 (training set) and 2023 (test set) (bottom two panels). In the top panels, red dashed lines indicate the empirical exceedance probabilities. In the bottom panels, red points denote the observed total values.

5.4.3 Covariates and latent effects

XGBoost model interpretation

The XGBoost model provides point forecasts of wildfire activity, while the latent Gaussian model primarily quantifies associated uncertainty. It is therefore essential to understand which covariates most strongly influence the predictions generated by XGBoost. To this end, we assess covariate importance using SHapley Additive exPlanations (SHAP) values (Lundberg and Lee, 2017).

SHAP values offer a principled approach to attributing the marginal contribution of each covariate to the model output, drawing on the concept of Shapley values from cooperative game theory (Shapley, 1953). For a model f fitted on a covariate set $M = \{\widetilde{x}_1, \widetilde{x}_2, \cdots, \widetilde{x}_m\}$, and a subset $S \subseteq M$, the SHAP value ϕ_j of covariate $\widetilde{x}_j$ is the average difference between the predictions $f(S \cup \{\widetilde{x}_j\})$ and $f(S)$ over all possible S. Formally, ϕ_j is defined by

$$\phi_j = \sum_{S \subseteq M \setminus \{\widetilde{x}_j\}} \frac{|S|!(|M| - |S| - 1)!}{|M|!} \left[f(S \cup \{\widetilde{x}_j\}) - f(S) \right].$$

Since f typically requires the full covariate set M, the output for a reduced set S is approximated as $f(S) = \mathbb{E}[f(M) \mid S]$. For tree-based models such as XGBoost, this conditional expectation can be efficiently computed using the algorithm proposed by Lundberg et al. (2018).

A key property of SHAP values is that they enable an additive decomposition of the model prediction:

$$f(M) = \mathbb{E}(f(M)) + \sum_{j=1}^{m} \phi_j. \tag{5.13}$$

Lundberg and Lee (2017) showed that (5.13) is the unique additive representation that satisfies local accuracy, missingness, and consistency. This decomposition provides an intuitive and theoretically grounded measure of covariate influence, based on both the sign and magnitude of the SHAP values.

Figure 5.8 displays the SHAP values for the ten most influential covariates in the XGBoost models for fire count (5.1) and burnt area (5.2). In both models, autoregressive terms dominate the set of top covariates, suggesting that historical wildfire activity contributes more to predictive accuracy than the environmental variables. The most influential covariates are the averages of fire count and burnt area of the three months centred at the forecast month over the past three years, aggregated at the council level (`conc_fc_hist_3` and `conc_ba_hist_3`, respectively). While high values of these autoregressive covariates do not always lead to large forecasts, they are generally positively correlated with wildfire events. Among the environmental covariates, average air temperature (`Temp`) contributes most strongly to the fire count model, while both `Temp` and relative humidity (`RHumi`) are most important for the burnt area model. The SHAP values of these variables exhibit an intuitive, non-causal relationship with the wildfire activity. For instance, high fire count and burnt area are often associated with elevated temperatures (reflected by large positive SHAP values), whereas low fire activity can occur across a broader range of temperature levels (indicated by the mix of blue and red at the lower end of the SHAP value), consistent with domain expectations.

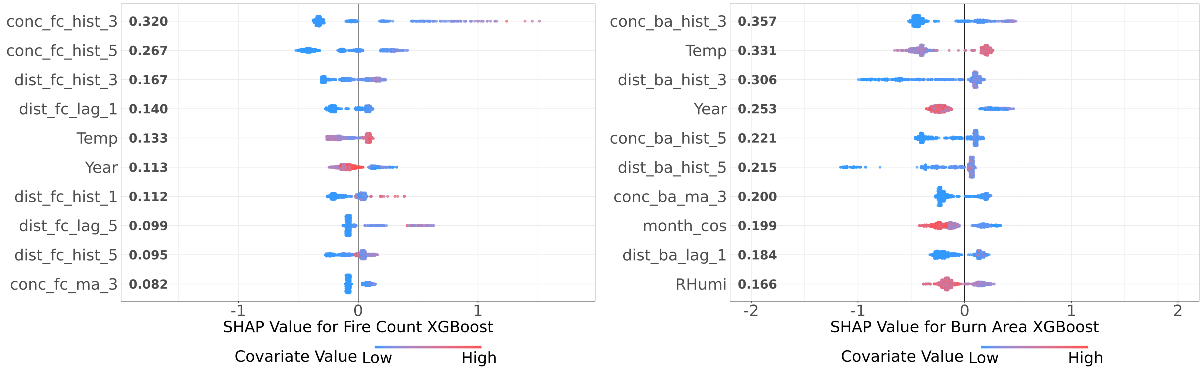

Figure 5.8: SHAP values for the top 10 covariates in the XGBoost models for fire count and burnt area. Covariates are ranked by the mean absolute SHAP value (numbers next to the covariate names) across all predictions. Full descriptions of the covariates are provided in Table 5.3 and 5.4.

Latent Gaussian effect of year and covariates

Figure 5.9 shows the posterior estimates of the year-specific effects $T(\cdot)$ and the effects of XGBoost predictions $R(\cdot)$ in η^C and η^B as specified in (5.7). While the year effects vary over time, only those for fire presence in 2017 and 2018 are significantly different from zero. This aligns with historical records: Portugal experienced the highest number of fire ignitions in 2017 in the past decade, followed by a sharp decline in 2018, possibly due to fire regulation adjustments after the catastrophic wildfire activity of 2017. As the XGBoost model already incorporates temporal information, the absence of significant year effects in most other years suggests that its predictions capture interannual variation effectively.

The lower panels of Figure 5.9 illustrate the effects of the XGBoost predictions for fire count and burnt area within the linear predictors η^C and η^B, respectively. In both cases, a generally increasing relationship is observed, indicating that higher XGBoost predictions are associated with higher contributions to the linear predictor. However, the relationship is not strictly linear, especially in the case of large burnt area values. This nonlinearity may stem from the differing likelihood assumptions in the two modelling components: the XGBoost model assumes a Gamma distribution (coming from the Tweedie loss) for positive burnt areas, while the latent Gaussian model employs an extended Generalised Pareto (eGP) distribution.

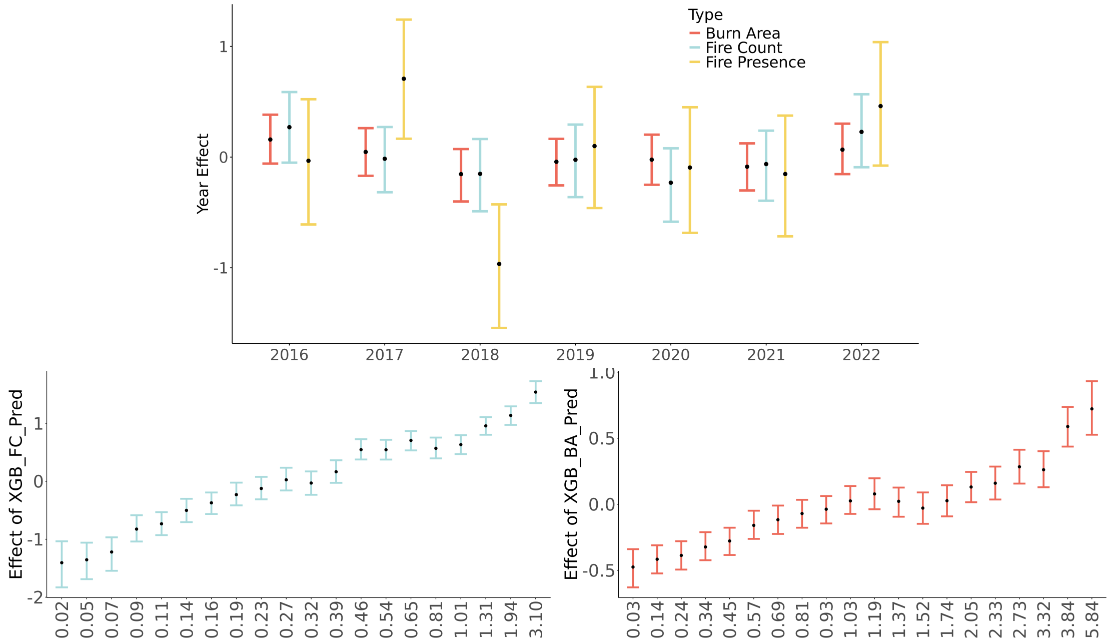

Figure 5.9: Posterior estimation of year effect $T(\cdot)$ (top) and XGBoost prediction effects $R(\cdot)$ in η^C (bottom left) and η^B (bottom right). The values on the x-axis in the bottom two correspond to the raw XGBoost forecasts at each spatio-temporal unit s, t. Black points in the three panels represent posterior means, while the vertical bars show 95% credible intervals.

Shared spatio-temporal effects

Figure 5.10 displays the posterior means of the council-level spatio-temporal effect $G_c(\cdot)$, grouped by month, and the average district-level spatio-temporal effect $G_d(\cdot)$ across all time indices. Their uncertainty, quantified by 0.025 and 0.975 posterior quantiles, is provided in Figure 5.14. During the high-risk wildfire season, particularly in July, August, and October, the council-level effects exhibit greater spatial variability, with notable contrasts between neighbouring councils. In contrast, during the remaining months, the spatial effects are more homogeneous and show smooth transitions across adjacent regions. Noteworthy exceptions include *Montalegre* and *Vinhais* councils: the former shows unusually elevated effects in January and March, while the latter displays pronounced effects in April relative to its surrounding areas. At the district level,

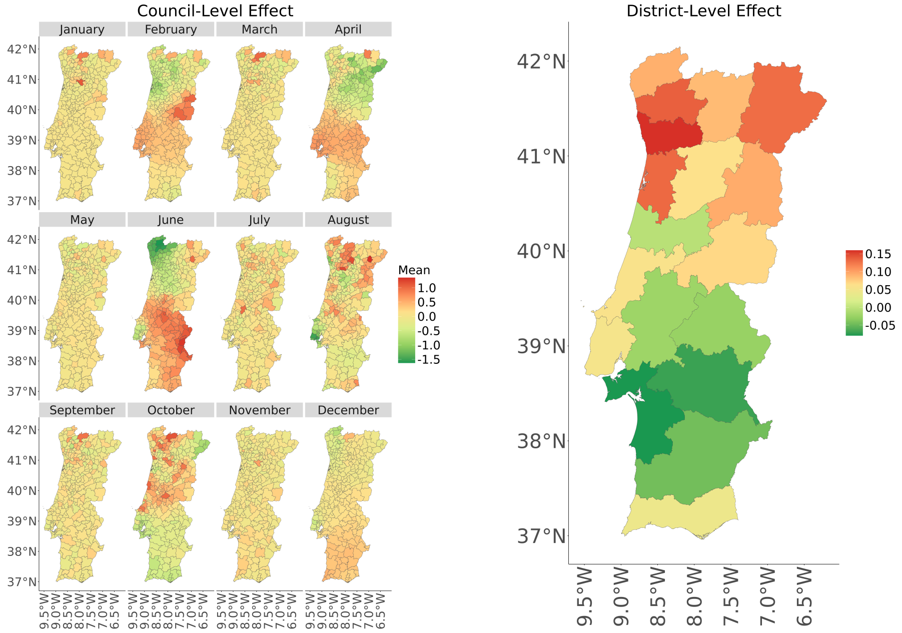

Figure 5.10: Posterior mean of the council-level spatio-temporal effect $G_c(\cdot)$ by grouped month (left), and average of the posterior mean of the district-level spatio-temporal effect $G_d(\cdot)$ (right), aggregated over all time indices.

a general spatial gradient is evident, with higher effect values observed in the northern districts and lower values in the south. However, the scale of the average district level is relatively small, with the maximum magnitude reaching only about 15% of that in the council-level.

The estimated posterior means of the scaling parameters for the shared effects are $\widehat{\beta}_1^C = -0.69$, $\widehat{\beta}_1^B = -0.43$, $\widehat{\beta}_2^C = 0.77$, $\widehat{\beta}_2^B = 0.27$. The signs of these parameters are not straightforward to interpret, given the dependence between council- and district-level spatio-temporal effects. Focusing instead on magnitudes within each wildfire quantity, the scaling parameters for fire count are consistently larger than those for burnt area, suggesting that the shared spatio-temporal effects for fire count are more strongly correlated with those for fire presence. This is because fire count is obtained by aggregating fire presence events, whereas burnt area is a continuous measure conditional on those events.

5.4.4 Posterior Distributions of eGP Parameters

We now provide insights into the posterior inference for the parameters ξ and κ from the perspective of implementing the eGP likelihood within the INLA framework. These parameters are treated as global hyperparameters in the latent Gaussian model, meaning they are shared across all observations. Figure 5.11 shows the prior and posterior distributions for ξ and κ. The posterior of ξ, compared to its prior centred around zero, shifts markedly to the right and concentrates near 0.45. This suggests that the fitted eGP likelihood possesses a heavy-tailed structure, which may be suitable for capturing the extreme behaviour observed in burnt area data. Notably, the mode of the posterior for ξ lies close to the upper bound of its prior, potentially conflicting with the imposed constraint of $(-0.5, 0.5)$. However, we argue that the posterior of ξ is particularly sensitive to the skewness and overall shape of the response distribution, and that its value has only a minor influence on the posterior predictive distribution of burnt area. This argument is further examined in Section 5.5.2. Therefore, rather than focusing on the interpretability of the posterior of ξ, it is more critical to ensure that the fitted eGP likelihood retains desirable properties, such as finite variance, by appropriately constraining the prior. As for κ, its posterior distribution deviates substantially from the prior, with mode concentrated around 4.6, though the posterior of κ shows larger dispersion compared to the posterior of ξ. This indicates that the posterior predictive probability $f_{\text{eGP}}(y_{s,t+1} \mid \eta^B_{s,t+1}, \xi, \kappa)$ does not have a singularity at 0, and its shape is approximately bell-like, though potentially skewed (see Figure 5.12).

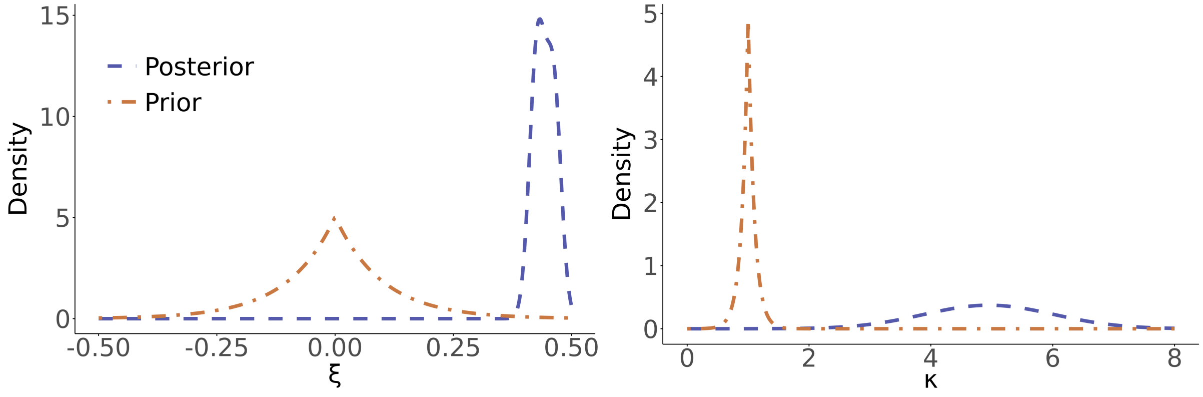

Figure 5.11: Prior and posterior distributions of the hyperparameters ξ (left) and κ (right) in the eGP likelihood. The prior for ξ is $\pi(\xi) = 10\exp\{-10|\xi|\}/(2-2\exp\{-5\}), \quad -0.5 < \xi < 0.5$, and the prior for κ is $\pi(\kappa) = \pi_2(\kappa) \; = 10\exp(-10|\kappa-1|)/(2-\exp(-10)), \quad \kappa > 0$.

5.5 Discussion

5.5.1 Data transformation and ξ in eGP

In our framework, the burnt area is modelled on the square root scale, rather than on its original scale or under alternative transformations. This decision is guided by both theoretical and empirical considerations. The square root transformation reduces the extreme skewness of the burnt area distribution while preserving a meaningful distinction between small and large events.

This results in a more stable fit of the extended Generalised Pareto (eGP) model across the full range of the data. In particular, the eGP tail parameter ξ is sensitive to the shape of the distribution in both the bulk and the tail. When working on the original scale, the strong skewness of burnt area data leads the model to prioritise fitting the bulk, often inflating tail estimates ($\xi > 0.5$), and causing tension with the prior support. On the other hand, aggressive transformations such as the logarithm overly compress the upper tail, causing ξ to collapse toward the lower bound of its prior support (-0.5), which in turn can distort inference about extremes.

The square root transformation strikes a practical balance, moderating skewness without unduly suppressing large values. Our empirical investigations, which are guided by posterior predictive checks and sensitivity analyses, show that this transformation yields stable and interpretable posteriors for ξ, consistent with prior beliefs and with tail behaviour observed in historical burnt area data. This choice aligns with modelling choices in recent wildfire literature, such as Cisneros et al. (2024). While alternative power transformations (e.g., cube root, fourth root, or logarithm) influence the posterior of ξ, they result in nearly identical posterior predictive distributions once back-transformed, reinforcing the square root as a pragmatic and robust choice.

5.5.2 Similar performance to Gamma and Weibull likelihoods

Although the eGP likelihood outperforms the Gamma and Weibull likelihoods in Table 5.1, the differences are modest, particularly relative to the Weibull, which also exhibits heavy-tailed behaviour with a posterior mean shape parameter of approximately 1.35. Figure 5.12 compares the three likelihoods using estimated hyperparameters. The linear predictors have been adjusted so that the medians of the resulting distributions align at approximately 0.8, facilitating a shape comparison. While the eGP distribution displays a heavier tail than the other two, the overall shapes of the three densities are broadly similar, with the largest differences appearing near the mode. These findings, consistent with Table 5.1, indicate that refining the likelihood to capture tail behaviour has limited impact on forecast performance; instead, accuracy is primarily determined by the flexibility of the latent structure and the incorporation of informative covariates into the linear predictor. Indeed, within the hierarchical structure of our model, observations are assumed conditionally independent given their respective linear predictors, which are functions of Gaussian latent effects. The central tendency and quantiles of the marginal distribution (e.g., its mean or α-quantile) are controlled by the linear predictor. Since hyperparameters such as ξ and κ are shared across all observations, their influence on individual predictive densities is limited relative to the localised effect of the linear predictor. When the latent structure is sufficiently expressive, especially through the inclusion of informative covariates such as the XGBoost predictions, the linear predictor can effectively account for both moderate and extreme observations. Consequently, the observations tend to cluster within the high-density region of the fitted marginal distribution, diminishing the role of the tail parameter in shaping the likelihood. As a result, the contribution of tail behaviour to overall model performance becomes less critical, which explains the comparable predictive accuracy across the three likelihoods.

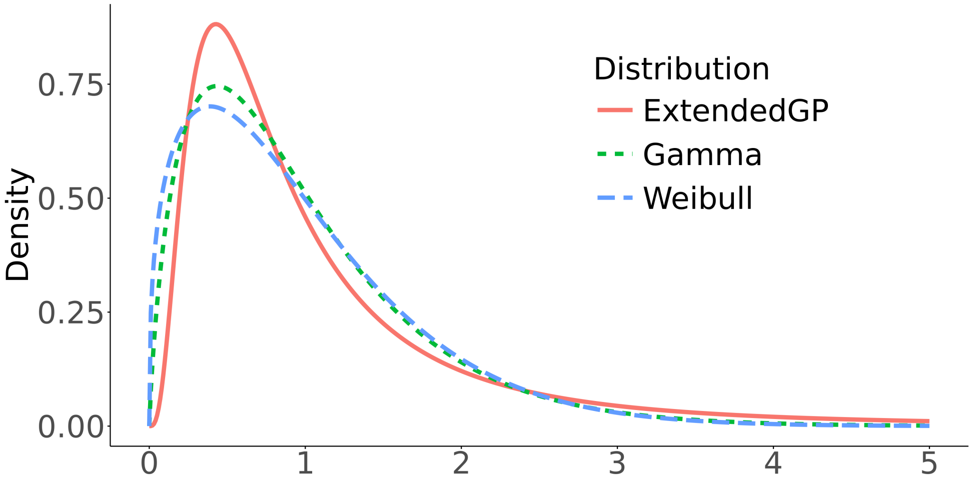

Figure 5.12: Densities of Gamma (shape = 1.80), Weibull (shape = 1.35) and eGP ($\kappa = 4.64$, $\xi = 0.45$) likelihoods, based on estimated hyperparameters. The densities are scaled via the linear predictor to have approximately equal medians (around 0.8).

5.5.3 Longer forecasting horizons

The current forecasting horizon considered in our framework is one month. That is, we generate forecasts y_{t+1} conditional on the past observations $y_{1:t}$ and additional covariates $\widetilde{\boldsymbol{x}}_t$. Extending this to longer-term forecasts (e.g. forecasting y_{t+h} for $h > 1$) presents a key challenge: neither the XGBoost model nor the latent Gaussian model is inherently designed to produce multi-horizon predictions in a unified structure. As such, it is not straightforward to generate forecasts across multiple future time points using a single model fit.

A practical solution is to apply the full two-stage modelling framework (as illustrated in Figure 5.3) independently for each forecasting horizon h. While the process has been detailed for $h = 1$, the extension to $h = 2, 3, \ldots$ involves adjusting and retraining the XGBoost model to predict future outcomes based on features lagged by h months. We emphasise that this approach does not constitute a fully coherent joint multi-horizon forecasting model, but rather a pragmatic, horizon-specific solution that remains compatible with the INLA framework and preserves interpretability and uncertainty quantification. Specifically, the XGBoost models can be modified to produce:

$$\begin{aligned} \widehat{y}^{C}_{s,t+h} &= \sum_{m} f^{C}_{m}(\widetilde{\boldsymbol{x}}^{C}_{s,t}), \\ \sqrt{\widehat{y}^{B}_{s,t+h}} &= \sum_{m} f^{B}_{m}(\widetilde{\boldsymbol{x}}^{B}_{s,t}), \end{aligned}$$

where $\widetilde{x}^{(\cdot)}_{s,t}$ denotes the feature vectors at space s and time t for fire count (C) and burnt area (B). These yield point forecasts of fire count and burnt area at time $t + h$, conditional on information available at time t. These forecasts then serve as inputs to the latent Gaussian model in the second stage, which is used to generate probabilistic predictions for the target variables at the specified horizon. This iterative, horizon-specific approach offers a tractable solution for multi-step forecasting while preserving the interpretability and modularity of the original model structure.

Since the most influential covariates in the XGBoost models for $h = 1$ are derived from historical wildfire activity at the same forecast month (See Figure 5.8 and Table 5.4), these predictors tend to continue to dominate the feature set when $h > 1$. Consequently, the predictive accuracy of the XGBoost models does not deteriorate substantially for longer forecast horizons ($h > 1$). This stability implies that the proposed multi-horizon forecast method is less susceptible to error propagation over time and can deliver robust uncertainty quantification. Table 5.2 presents a comprehensive comparison of forecast accuracy and uncertainty for model M1 with the eGP likelihood across horizons $h = 1, 2, 3$. In Stage 1, mean squared error (MSE) is used to assess the performance of the XGBoost models. In Stage 2, forecast accuracy is evaluated using the same metrics as in Table 5.1, while uncertainty is quantified by the average length of the 95% credible interval derived from INLA. Only minor variations in predictive performance and uncertainty are observed across horizons from one to three months, with no evidence of a systematic degradation in performance.

5.5.4 Choices of the model in stage 1

In this study, XGBoost is adopted in Stage 1 to generate one-month-ahead wildfire forecasts at the council level. Since Stage 1 only requires point forecasts, XGBoost is not the only viable choice. Alternative models, such as ensemble methods (e.g., Random Forest; Breiman, 2001)

Table 5.2: Comparison of prediction accuracy and uncertainty for model M1 across forecast horizons $h = 1, 2, 3$. Mean squared error (MSE) evaluates the accuracy of the two Stage 1 XGBoost models. Stage 2 prediction metrics are identical to those reported in Table 5.1. Uncertainty in Stage 2 is quantified by the average length of the 95% credible interval.

Forecast Horizon		h=1	h=2	h=3
Stage 1 Prediction	MSE for count	0.34	0.36	0.30
	MSE for burnt area	8.26	8.27	8.24
Stage 2 Prediction	AUC	0.862	0.857	0.864
	CRPS	4.75	4.61	4.61
	Unweighted r^C	362	367	345
	Weighted r^C	4.84	5.11	4.83
	Unweighted r^B	565	585	599
	Weighted r^B	14.2	14.8	14.4
Stage 2 Uncertainty	Avg. CI for count	3.38	3.49	3.50
	Avg. CI for burnt area	24.33	24.28	23.92

or deep learning approaches for time series forecasting (e.g., long short-term memory networks; Hochreiter and Schmidhuber, 1997, and attention-based models; Lim et al., 2021; Zhou et al., 2021), could serve a similar role. In principle, multiple models targeting the same response could be fitted in Stage 1, with their outputs jointly incorporated as covariates in Stage 2. This strategy is not pursued here because the number of hyperparameters associated with existing model components is already close to 20, which is near the recommended upper limit for INLA. When selecting a single model, numerous studies have demonstrated that XGBoost achieves state-of-the-art performance in both time series forecasting (Elsayed et al., 2021; Januschowski et al., 2022; Santoro et al., 2024) and tabular data modelling (Gorishniy et al., 2021; Grinsztajn et al., 2022). For these reasons, XGBoost is chosen as the Stage 1 model in our framework.

5.6 Conclusion

In this study, we propose a two-stage ensemble modelling framework that addresses the challenge of incorporating multiple future covariates in spatio-temporal forecasting using INLA. Our approach integrates window-based XGBoost predictions as proxy covariates for future fire count and burnt area, and couples them with a latent Gaussian model to produce calibrated posterior forecasts. By compressing complex spatio-temporal dynamics and dozens of environmental predictors into two informative proxy covariates, the framework enables INLA to exploit information that would otherwise be difficult to accommodate, thereby improving predictive performance.

Furthermore, we introduce and implement the novel sub-asymptotic eGP likelihood within the INLA framework and its companion R-INLA library, enabling joint modelling of both moderate and extreme wildfire events.

By comparing posterior predictions under the eGP, Weibull, and Gamma likelihoods while keeping the remainder of the model structure fixed, we observe only a marginal improvement in predictive performance with the eGP likelihood. We attribute this to the conditional independence assumption and the dominance of the linear predictor in shaping the marginal likelihoods. Similar findings are reported in Yadav et al. (2023), who also reported minimal sensitivity to likelihood choice in Bayesian hierarchical models for landslide size.

We also discuss a strategy for extending the framework to longer forecasting horizons by replicating the two-stage procedure for each horizon separately. While this does not offer a unified multi-horizon forecast, it provides a practical path forward within the current model constraints. The result shows that our proposed framework can provide a stable forecast and uncertainty for longer horizons.

One remaining limitation of the present work is the absence of explicit covariates that capture human activity, which is known to play a critical role in wildfire ignition and propagation. Incorporating such information, e.g., data on population density, land use, or proximity to infrastructure, could further enhance the predictive capacity of the model and is a promising direction for future research.

5.7 Supplementary material

5.7.1 Code and data

The code for implementing the two-stage model and reproducing the results in this paper is available at https://github.com/hcl516926907/Portugal_Wildfire. The wildfires and ERA5 data are publicly available online. Due to the size of the data, they are not shared in the above GitHub repository.

5.7.2 Derivation of the PC prior for $\kappa > 0$

We construct a penalised complexity (PC) prior for the parameter κ in the extended generalised Pareto (eGP) distribution. The base model corresponds to $\kappa = 1$, under which the eGP reduces to the standard Generalised Pareto distribution (GPD).

Let $f_\kappa(y) \equiv f_{\text{eGP}}(y;\kappa)$ and $f_{\kappa_1}(y) \equiv f_{\text{eGP}}(y;\kappa=1)$ denote the eGP densities for general $\kappa > 0$ and for the base model, respectively. The PC prior is defined by assigning an exponential distribution to the Kullback–Leibler-based distance between f_κ and the base model:

$$d(\kappa) = \sqrt{2\,\text{KLD}(f_\kappa \| f_{\kappa_1})}.$$

This distance quantifies the additional complexity introduced by allowing $\kappa \neq 1$. The PC prior is then given by:

$$\pi(\kappa) = \lambda \exp\{-\lambda d(\kappa)\} \left| \frac{\partial d(\kappa)}{\partial \kappa} \right|,$$

where $\lambda > 0$ is a user-defined rate parameter.

Case $\boldsymbol{\xi} \neq \mathbf{0}$. The Kullback–Leibler divergence from f_{κ_1} to f_κ is defined as:

$$\text{KLD}(f_\kappa \| f_{\kappa_1}) = \int_0^\infty f_\kappa(y) \log\left(\frac{f_\kappa(y)}{f_{\kappa_1}(y)}\right) \mathrm{d}y.$$

Using the expression for the eGP density when $\xi \neq 0$, we have:

$$\begin{aligned} f_\kappa(y) &= \kappa \left[1 - \left(1 + \xi \frac{y}{\sigma}\right)^{-\frac{1}{\xi}}\right]^{\kappa-1} \cdot \frac{\xi}{\sigma}\left(1 + \xi\frac{y}{\sigma}\right)^{-\frac{1}{\xi}-1}, \\ f_{\kappa_1}(y) &= \frac{\xi}{\sigma}\left(1 + \xi\frac{y}{\sigma}\right)^{-\frac{1}{\xi}-1}. \end{aligned}$$

Hence,

$$\log\left(\frac{f_\kappa(y)}{f_{\kappa_1}(y)}\right) = \log(\kappa) + (\kappa - 1)\log\left(1 - \left(1 + \xi\frac{y}{\sigma}\right)^{-\frac{1}{\xi}}\right),$$

and the KLD becomes:

$$\int_0^\infty f_\kappa(y) \cdot \log\left(\frac{f_\kappa(y)}{f_{\kappa_1}(y)}\right) \mathrm{d}y = \log(\kappa)\int_0^\infty f_\kappa(y)\,\mathrm{d}y + (\kappa-1)\int_0^\infty f_\kappa(y)\log\left(1 - \left(1 + \xi\frac{y}{\sigma}\right)^{-\frac{1}{\xi}}\right)\mathrm{d}y. \tag{5.14}$$

The first term in (5.14) simplifies directly in $\log\kappa$, since $\int_0^\infty f_\kappa(y)\mathrm{d}y = 1$. For the second term, applying the change of variable $u = 1 - (1 + \xi y/\sigma)^{-1/\xi}$, yielding:

$$(\kappa - 1)\int_0^\infty f_\kappa(y)\log\left(1 - \left(1 + \xi\frac{y}{\sigma}\right)^{-\frac{1}{\xi}}\right)\mathrm{d}y = (\kappa-1)\int_0^1 \kappa u^{\kappa-1}\log u\,\mathrm{d}u = -\frac{\kappa-1}{\kappa}$$

Therefore,

$$\text{KLD}(f_\kappa \| f_{\kappa_1}) = \log\kappa - \frac{\kappa - 1}{\kappa}. \tag{5.15}$$

Using the exact KLD in (5.15), the corresponding PC prior is:

$$\pi(\kappa) = \begin{cases} \frac{\lambda|\kappa-1|}{2\kappa^2\sqrt{2\log\kappa - 2(\kappa-1)/\kappa}} \exp\left\{-\lambda\left(\sqrt{2\log\kappa - 2(\kappa-1)/\kappa}\right)\right\}, & \kappa > 0,\ \kappa \neq 1, \\ \lambda/2, & \kappa = 1. \end{cases}$$

Alternatively, approximating the KLD around $\kappa = 1$ via a second-order Taylor expansion yields:

$$\text{KLD}(f_\kappa \| f_{\kappa_1}) = \frac{1}{2}(\kappa-1)^2 + o((\kappa-1)^3), \qquad \text{as } \kappa \to 1.$$

This leads to a locally symmetric PC prior:

$$\pi(\kappa) \ = \frac{\lambda\exp(-\lambda|\kappa-1|)}{2-\exp(-\lambda)}, \qquad \kappa > 0.$$

Case $\boldsymbol{\xi = 0}$. When $\xi = 0$, the eGP reduces to a power transformation of the exponential distribution:

$$f_\kappa(y) = \kappa\left(1 - \exp\left\{-\frac{y}{\sigma}\right\}\right)^{\kappa-1} \cdot \frac{1}{\sigma}\exp\left\{-\frac{y}{\sigma}\right\},$$
$$f_{\kappa_1}(y) = \frac{1}{\sigma}\exp\left\{-\frac{y}{\sigma}\right\}.$$

Using the change of variable $v = 1 - \exp\{-y/\sigma\}$, one can derive the same KLD expression as in the case $\xi \neq 0$, thereby recovering the same PC prior $\pi(\kappa)$.

5.7.3 Covariates in XGBoost

Table 5.3 provides a description of the environmental covariates derived from the ERA5 dataset, while Table 5.4 summarises the feature-engineered autoregressive covariates constructed from historical wildfire records.

Table 5.3: Environmental Covariates used in the XGBoost model.

Name	**Source**	**Spatial Resolution**	**Temporal Resolution**	**Description**
Pricp	ERA5-Land	$0.1° \times 0.1°$	Hourly	Accumulated liquid and frozen water, including rain and snow, that falls to the Earth's surface.
Temp	ERA5-Land	$0.1° \times 0.1°$	Hourly	Temperature of air at 2m above the surface of land.
Ucomp	ERA5-Land	$0.1° \times 0.1°$	Hourly	Eastward component of the 10m wind.
Vcomp	ERA5-Land	$0.1° \times 0.1°$	Hourly	Northward component of the 10m wind.
DewPoint	ERA5-Land	$0.1° \times 0.1°$	Hourly	Temperature to which the air, at 2 metres above the surface of the Earth, would have to be cooled for saturation to occur.
HVegLAI	ERA5-Land	$0.1° \times 0.1°$	Hourly	One-half of the total green leaf area per unit horizontal ground surface area for high vegetation type.
LVegLAI	ERA5-Land	$0.1° \times 0.1°$	Hourly	One-half of the total green leaf area per unit. horizontal ground surface area for low vegetation type.
HVegCov	ERA5	$0.25° \times 0.25°$	Constant	The fraction of the grid box that is covered with vegetation that is classified as "high".
LVegCov	ERA5	$0.25° \times 0.25°$	Constant	The fraction of the grid box that is covered with vegetation that is classified as "low".
HVegTyp	ERA5	$0.25° \times 0.25°$	Constant	Indicator of the 6 types of high vegetation recognised by the ECMWF Integrated Forecasting System.
LVegTyp	ERA5	$0.25° \times 0.25°$	Constant	Indicator of the 10 types of low vegetation recognised by the ECMWF Integrated Forecasting System.
FWI	Derived	$0.1° \times 0.1°$	Hourly	Fire Weather Index
RHumi	Derived	$0.1° \times 0.1°$	Hourly	Relative Humidity

Table 5.4: Feature-engineered autoregressive covariates for fire count and burnt area. In this table, t denotes the forecasting time point, h is the forecast horizon, X indicates the spatial scale, taking values "dist" (district level) or "conc" (council level); and Y specifies the source variable, with "fc" for fire count and "ba" for burnt area.

Name	Index Range	Formula	Description
X_Y_lag_j	$j = 1, 2, \cdots, 9$	$y_{s,t-j}$	Lag j of monthly fire count/burnt area at council/district level
X_Y_ma_j	$j = 3, 6, 9, 12, 24, 36$	$\frac{\sum_{i=1}^{j} y_{s,t-i}}{j}$	Moving average of past j months of fire count/burnt area at council/district level
X_Y_hist_j	$j = 1, 3, 5$	$\frac{\sum_{k=1}^{3} \sum_{i=-(j-1)/2}^{(j-1)/2} y_{s,t+h-12k+i}}{3j}$	Average fire count/burnt area over centred j month around month $t + h$ in the past 3 years
month_sin	NA	$\sin(2\pi t/12)$	Angular representation of the month
month_cos	NA	$\cos(2\pi t/12)$	Angular representation of the month
Year	NA	NA	Year of the wildfire occurrence
Lon	NA	NA	Longitude of the centroid of the council
Lat	NA	NA	Latitude of the centroid of the council

5.7.4 Additional diagnostic plots

We use the ACF plots in Figure 5.13 to determine the extent of long-term temporal dependence, specifically, the number of lags or amount of historical data to include in constructing the autoregressive covariates. For each month, we first calculate the average fire count and burnt area across all councils, and then compute the autocorrelation using these monthly averages following the standard ACF formula.

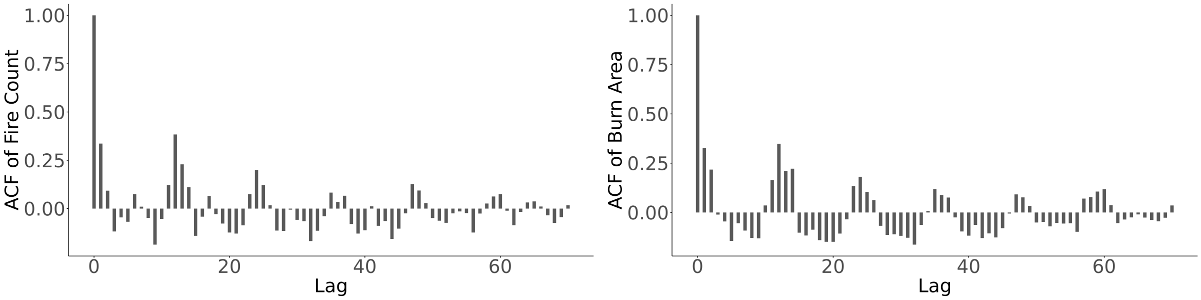

Figure 5.13: Autocorrelation Function plots of average fire count and burnt area at council-monthly level.

The uncertainty associated with the council-level spatio-temporal effect $G_c(\cdot)$ and the district-level spatio-temporal effect $G_d(\cdot)$ is illustrated below. Uncertainty is quantified using the 0.025 and 0.975 posterior quantiles of the corresponding random effects.

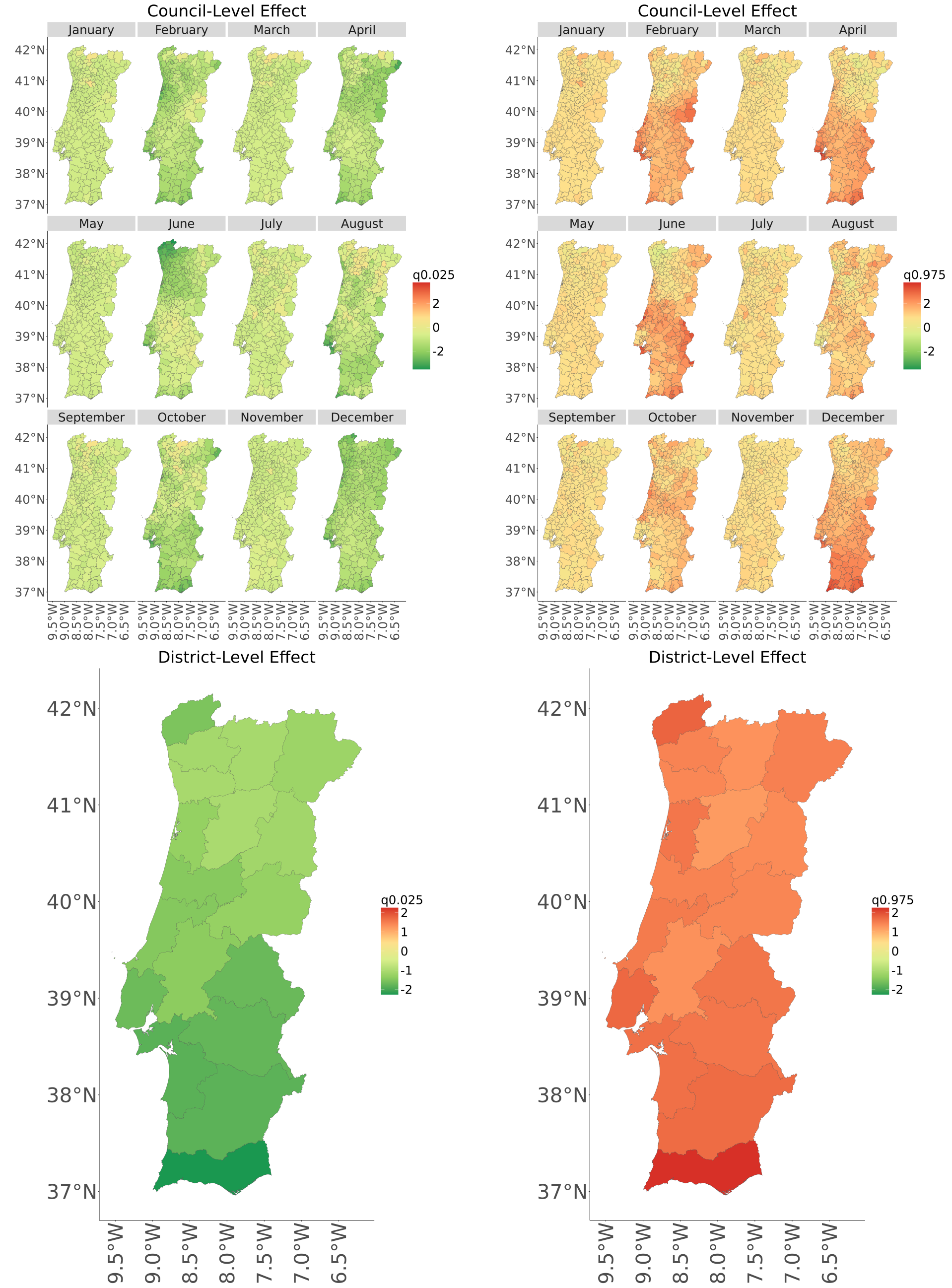

Figure 5.14: Posterior uncertainty of the spatio-temporal effects. The top panel shows the 0.025 and 0.975 posterior quantiles of the council-level spatio-temporal effect $G_c(\cdot)$ grouped by month, while the bottom panel displays the averages of the 0.025 and 0.975 posterior quantiles of the district-level spatio-temporal effect $G_d(\cdot)$ aggregated over all time indices.

Chapter 6

Modelling bulk and tail distributions with deep learning

Estimating probabilities of simultaneous high-threshold exceedances is central for assessing joint risk from extremes (e.g., precipitation). While multivariate extreme value theory (EVT) often relies on max-stability, this assumption can be overly restrictive for environmental data and is rarely tenable in high dimensions. Motivated by the Extreme Value Data Challenge 2025, we adopt a purely deep generative approach for extreme precipitation on a 5-by-5 grid (25 sites) using 60,225 daily observations from the CESM2 Large Ensemble (LENS2), corresponding to 165 simulated years with geographic information removed. We model temporal dependence with a Long Short-Term Memory network and, conditional on its output, use a diffusion model to learn the joint daily precipitation distribution across sites. To handle heavy tails and zero inflation, we apply a customised logarithmic transform. We evaluate performance via (i) exceedance probabilities at six high thresholds, (ii) marginal quantiles, and (iii) marginal tail-shape parameters from annual block maxima. Across metrics, the proposed model accurately captures marginal tails and joint tail behaviour for daily precipitation.

6.1 Introduction

Quantifying the risk of extreme precipitation is fundamental in environmental science, with direct relevance to hazards such as flooding and landslides (Iverson, 2000; Westra et al., 2014). For a single site, EVT provides principled tools, including block maxima with a GEV fit (Alaya et al., 2020; Tabari, 2021) and peaks-over-threshold based on the GPD (Wi et al., 2016;

Thiombiano et al., 2017). While these univariate models rely on max-stability as a foundational limit assumption, extending the framework to multiple sites is substantially more challenging: classical multivariate EVT is typically built on *joint* max-stability (e.g., multivariate GEV (Naveau and Segers, 2024)), an assumption that often becomes unrealistic in environmental applications and increasingly restrictive in higher dimensions (Huser et al., 2025). Key issues include:

1. **Restrictive tail dependence**: max-stable models enforce asymptotic dependence, potentially biasing inference when the data are asymptotically independent. This limitation arises from the fact that multivariate max-stable distributions are fundamentally constructed on component-wise maxima, which can over-emphasise simultaneous extremes across sites (Beirlant et al., 2006).

2. **Limited flexibility**: even when restricting to asymptotically dependent cases, multivariate max-stable models lack flexibility due to the infinite-dimensional nature of their dependence structures. In practice, only a limited subclass of parametric forms is typically used (Coles et al., 2001; Rootzén and Tajvidi, 2006).

A range of alternatives has been developed, including conditional extremes (Heffernan and Tawn, 2004), geometric extremes (Wadsworth and Campbell, 2024; Murphy-Barltrop et al., 2024; De Monte et al., 2025), and angular–radial methods (Mackay et al., 2024, 2025). In parallel, recent work uses deep generative models for heavy-tailed data (Allouche et al., 2022; Hu and Castro-Camilo, 2025; Lhaut et al., 2025; De Monte et al., 2025).

In this paper, motivated by the Extreme Value Data Challenge 2025, we model daily precipitation on a 5-by-5 grid (25 sites) using Denoising Diffusion Probabilistic Models (Ho et al., 2020) with an emphasis on tail characterisation and extrapolation. We first apply a modified logarithmic transform to address zero inflation and heavy tails. Temporal dependence is captured with a Long Short-Term Memory network (LSTM, Hochreiter and Schmidhuber (1997)); conditional on the LSTM output, a diffusion model learns the joint distribution of transformed precipitation across the 25 sites. This structure adapts the autoregressive multivariate time-series framework of Rasul et al. (2021a) to an extremes setting.

As a baseline EVT approach, we model each site marginally by splicing a Weibull distribution below a high threshold with a GPD for exceedances; the marginal parameters are produced by the LSTM. The joint distribution across sites is then formed by assuming conditional independence given these parameters.

Overall, the diffusion model better captures heavy-tailed behaviour and yields more realistic tail dependence than the conditionally independent EVT baseline, though both approaches have difficulty extrapolating beyond the observed range.

The paper proceeds as follows. Section 6.2 briefly describes the data and challenge tasks, and summarises exploratory results on tail dependence. Section 6.3 presents the modelling frameworks. Section 6.4 compares models, and Section 6.5 discusses limitations and open challenges.

6.2 Data description and preliminary exploration

The dataset used in the Extreme Value Data Challenge 2025 comprises four simulated runs from the CESM2 Large Ensemble Community Project (LENS2). Each run contains daily precipitation records on a 5-by-5 grid spanning the period 1850–2014. Precipitation values are rescaled into an artificial unit called "Leadbetters", and, crucially, all geographical information about the grid has been removed. Because the locations cannot be interpreted spatially, we treat the 25 grid cells purely as components of a multivariate system. Thus, each day's observations are represented as a 25-dimensional vector, and the problem is framed as one of multivariate extreme-value modelling rather than spatial analysis.

The challenge requires estimating six specific quantities defined by the organisers, labelled Q_1 to Q_6, with two rounds of submission: a preliminary phase targeting Q_1 to Q_3, and a final phase, focusing on Q_4 to Q_6. These quantities are as follows:

Q_1. Expected number of times the sum of daily rainfall across all 25 locations exceeds 85 Leadbetters.

Q_2. Expected number of times at least 3 of the 5 sites in the first row exceed 4.3 Leadbetters.

Q_3. Expected number of times at least 3 of the 5 sites in the first row exceed 2.5 Leadbetters for a run of at least two consecutive days (not necessarily the same sites on both days; a run of three days counts only once).

Q_4. Expected number of times all 25 locations exceed 1.7 Leadbetters.

Q_5. Expected number of times at least 6 of the 25 sites exceed 5.7 Leadbetters.

Q_6. Expected number of times at least 3 of the 25 sites exceed 5 Leadbetters for a run of at least two consecutive days.

These quantities measure the expected number of various extreme event occurrences within each simulation run. Their mathematical formulations are provided in Section 6.4. To accurately estimate these quantities, it is essential to construct a model that captures the daily threshold exceedance probabilities while accounting for temporal dependence. Here, "threshold exceedance" refers specifically to partial threshold exceedance, i.e., cases in which at least one component exceeds a given threshold. This is critical, as the target quantities involve both joint extremes across all 25 sites and exceedances within subsets of the sites. In the latter case, exceedance probabilities must be computed by integrating over the full range of the non-extreme sites.

A natural starting point for modelling partial exceedances is the multivariate generalised Pareto distribution (mGPD) (Rootzén et al., 2018b; Kiriliouk et al., 2019), which is designed specifically for such tasks and has been extended to allow flexible dependence structures (Hu and Castro-Camilo, 2025). However, a fundamental limitation of the mGPD is that it inherits the max-stable assumption, implying asymptotic dependence regardless of the specified dependence structure. As a result, the extremal dependence metrics $\chi(q)$ and $\omega(q)$, defined as

$$\chi(q) = \frac{\mathbb{P}\{\bigcap_{j=1}^{25}\{X_j > F_j^{-1}(q)\}\}}{1-q}, \qquad \omega(q) = \frac{\mathbb{P}\{\bigcup_{j=1}^{25}\{X_j > F_j^{-1}(q)\}\}}{1-q}, \qquad q \in (0,1),$$

become a strictly positive constant for large enough q (Hu and Castro-Camilo, 2025), a characteristic feature of asymptotic dependence. Figure 6.1 shows empirical estimates of $\chi(q)$ and $\omega(q)$ using run1 computed separately for each calendar day across the years. The estimates show $\chi(q)$ decreasing and $\omega(q)$ increasing monotonically over the entire range $q \in (0.5, 0.99)$, with no indication of stabilisation. This strongly suggests that the max-stable assumption does not hold for this dataset, further highlighting the inadequacy of traditional multivariate EVT models in this context and reinforcing the need for alternative modelling approaches.

6.3 Methods

6.3.1 Long short-term memory

Long Short-Term Memory (LSTM) networks are a class of recurrent neural network (RNN) for sequential data. Given covariates $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$ and a multivariate series $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T)$, an LSTM updates a hidden state via

$$\boldsymbol{h}_t = f(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t; \boldsymbol{\theta}), \qquad t = 1, \ldots, T,$$

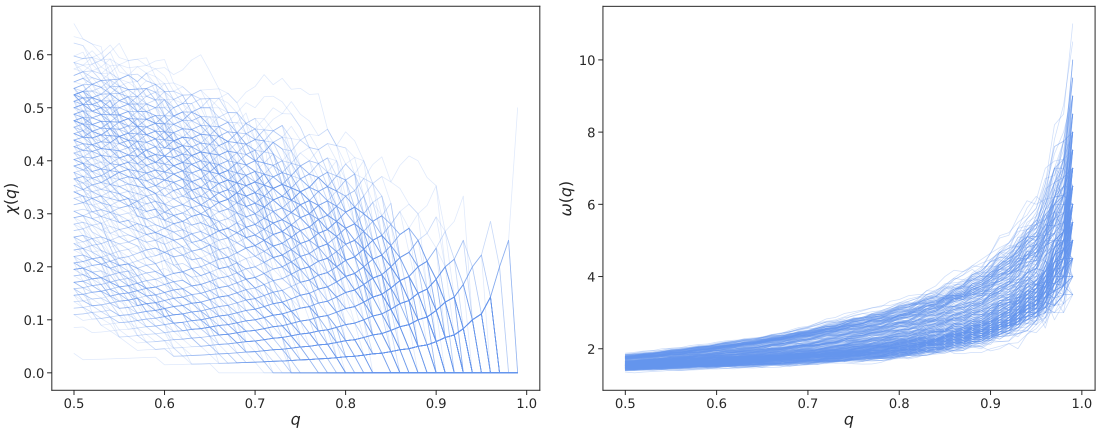

Figure 6.1: Tail dependence metrics $\chi(q)$ and $\omega(q)$ for run1, evaluated over $q \in (0.5, 0.99)$. Each of the 365 lines in each plot corresponds to the metric evaluated on a subset of run1 data for a given day of the year.

where f is a neural network design consisting of forget gates, input gates and output gates to efficiently update the historical memory and hidden state without the gradient exploding issue in traditional RNNs. The hidden state $\boldsymbol{h}_t$ summarises information up to time t and can be taken as the point forecast at t.

When a probabilistic forecast is needed, Salinas et al. (2020) proposed to specify an explicit likelihood for $\boldsymbol{y}_t \mid \boldsymbol{h}_t$ (often assuming conditional independence in high dimensions) and models its parameters as functions of $\boldsymbol{h}_t$. Such factorised likelihoods can under-represent cross-variable dependence. To improve flexibility, Rasul et al. (2021a,b) replace the conditional likelihood with a generative model (e.g., normalising flows or diffusion models) conditioned on $\boldsymbol{h}_t$. This hybrid approach, which integrates LSTM for temporal modelling and diffusion models for multivariate distribution estimation, forms the basis of the framework proposed in this study for extrapolating high-threshold exceedances in multivariate time series.

6.3.2 Diffusion models

We use Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020; Sohl-Dickstein et al., 2015) to approximate a data distribution $q(\boldsymbol{x}^0)$ by a tractable generative model $p_\theta(\boldsymbol{x}^0)$. DDPMs define (i) a *forward* noising Markov chain and (ii) a learned *reverse* denoising chain.

Forward process For noising step $n = 1, \ldots, N$, the latent state $\boldsymbol{x}^n$ has a state transition equation

$$q(\boldsymbol{x}^n \mid \boldsymbol{x}^{n-1}) = \mathcal{N}\left(\boldsymbol{x}^n; \sqrt{1-\beta_n}\, \boldsymbol{x}^{n-1}, \beta_n \boldsymbol{I}\right),$$

with a fixed noise schedule β_n. Writing $\alpha_n := 1 - \beta_n$ and $\bar{\alpha}_n := \prod_{i=1}^{n} \alpha_i$, the marginal has closed form

$$q(\boldsymbol{x}^n \mid \boldsymbol{x}^0) = \mathcal{N}\big(\boldsymbol{x}^n; \sqrt{\bar{\alpha}_n}\, \boldsymbol{x}^0, (1 - \bar{\alpha}_n)\boldsymbol{I}\big) \,,$$

so $\boldsymbol{x}^N$ approaches $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for large N. The joint forward density is $q(\boldsymbol{x}^{1:N} \mid \boldsymbol{x}^0) = \prod_{n=1}^{N} q(\boldsymbol{x}^n \mid \boldsymbol{x}^{n-1})$.

Reverse process and training The reverse chain starts from $p(\boldsymbol{x}^N) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and uses Gaussian transitions

$$p_\theta(\boldsymbol{x}^{n-1} \mid \boldsymbol{x}^n) = \mathcal{N}\big(\boldsymbol{x}^{n-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}^n, n), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}^n, n)\big) \,, \qquad p_\theta(\boldsymbol{x}^{0:N}) = p(\boldsymbol{x}^N) \prod_{n=1}^{N} p_\theta(\boldsymbol{x}^{n-1} \mid \boldsymbol{x}^n).$$

Following Ho et al. (2020), we fix $\boldsymbol{\Sigma}_\theta$ to be $\beta_n \boldsymbol{I}$ and parameterise the mean via a noise-prediction network $\boldsymbol{\epsilon}_\theta$,

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}^n, n) = \frac{1}{\sqrt{\alpha_n}} \left(\boldsymbol{x}^n - \frac{\beta_n}{\sqrt{1 - \bar{\alpha}_n}} \, \boldsymbol{\epsilon}_\theta(\boldsymbol{x}^n, n) \right) ,$$

and train $\boldsymbol{\epsilon}_\theta$ to minimise the following simplified objective derived from the evidence lower bound:

$$\mathbb{E}_{n, \boldsymbol{x}^0, \boldsymbol{\epsilon}} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_n} \boldsymbol{x}^0 + \sqrt{1 - \bar{\alpha}_n} \boldsymbol{\epsilon}, n) \right\|^2 \right], \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}).$$

Sampling After training, new sample $\boldsymbol{x}^0$ can be obtained by first drawing $\boldsymbol{x}^N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and iterate for $n = N, \ldots, 1$,

$$\boldsymbol{x}^{n-1} = \frac{1}{\sqrt{\alpha_n}} \left(\boldsymbol{x}^n - \frac{\beta_n}{\sqrt{1 - \bar{\alpha}_n}} \, \boldsymbol{\epsilon}_\theta(\boldsymbol{x}^n, n) \right) + \sigma_n \boldsymbol{z},$$

with $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for $n > 1$ and $\boldsymbol{z} = \mathbf{0}$ for $n = 1$.

6.3.3 Data transformation

Diffusion models assume continuous data, whereas precipitation is mixed: a point mass at zero plus continuous positive values. In addition, like other deep generative models, diffusion models can struggle with heavy-tailed targets (Pandey et al., 2024). We therefore apply a simple transformation that (i) removes discreteness near zero via jittering and (ii) reduces tail-heaviness via a log map.

For $\boldsymbol{y} \in \mathbb{R}_+^d$, define

$$T(\boldsymbol{y}) = \log\Big(\boldsymbol{u} \odot \boldsymbol{U} \odot \mathbb{1}(\boldsymbol{y} < \boldsymbol{u}) + \boldsymbol{y} \odot \mathbb{1}(\boldsymbol{y} \geq \boldsymbol{u})\Big),$$

with $\boldsymbol{u} = \mathbf{0.01}$. Thus, values below u_j are treated as effectively zero and replaced by a random draw in $(0, u_j)$ to yield a continuous input, after which we take logs. An (approximate) inverse map is

$$T^{-1}(\boldsymbol{x}) = \mathbf{0} \odot \mathbb{1}(\boldsymbol{x} < \log \boldsymbol{u}) + \exp\big(\boldsymbol{x} \odot \mathbb{1}(\boldsymbol{x} \geq \log \boldsymbol{u})\big).$$

Because of jittering, $T^{-1} \circ T$ is not exact for $y_j \in [0, 0.01)$, but the induced error is negligible for our focus on extremes.

A common EVT alternative is to fit a GPD above a high threshold and transform exceedances to (approximately) exponential margins. We do not use this here because (i) fixed GPD parameters implicitly assume stationarity, while the series is nonstationary across time and space. Fitting separate GPD models for each location and time block might partially address this, but it would require estimating hundreds of models, making it impractical. (ii) threshold-based transforms act only on the upper tail, whereas our latent-state framework requires a transformation applied to the full series. We therefore use the log-based transform, which is simple, robust to nonstationarity, and directly compatible with conditional generative modelling.

6.3.4 Latent state modelling

Let $d = 25$ and $T = 60{,}225$. For day t, let $\boldsymbol{y}_t \in \mathbb{R}^{25}$ denote precipitation and set $\boldsymbol{x}_t = T(\boldsymbol{y}_t)$. An LSTM first encodes $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_T)$ into hidden states $(\boldsymbol{h}_1, \dots, \boldsymbol{h}_T)$. Conditional on $\boldsymbol{h}_{t-1}$, the transformed observation is generated by a conditional diffusion model:

$$\boldsymbol{x}_t \sim p(\boldsymbol{x}_t \mid \boldsymbol{h}_{t-1}, \Theta), \qquad t = 2, \dots, T, \tag{6.1}$$

where Θ collects LSTM and diffusion parameters. The above structure is illustrated in Figure 6.2.

Under the Markov/conditional-independence assumption, we maximise the log-likelihood of the full sequence (excluding the initial day)

$$\log p(\boldsymbol{x}_{2:T} \mid \boldsymbol{h}_{1:(T-1)}, \Theta) = \sum_{t=2}^{T} \log p(\boldsymbol{x}_t \mid \boldsymbol{h}_{t-1}, \Theta),$$

where each term in the sum is trained via the conditional DDPM objective

$$\mathbb{E}_{n,\boldsymbol{x}_t,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_n}\boldsymbol{x}_t + \sqrt{1-\bar{\alpha}_n}\boldsymbol{\epsilon}, \boldsymbol{h}_{t-1}, n)\right\|^2\right]. \tag{6.2}$$

After training, we obtain $\widehat{\boldsymbol{h}}_{1:(T-1)}$ by running the observed $\boldsymbol{x}_{1:(T-1)}$ through the LSTM, then sample $\widetilde{\boldsymbol{x}}_t \sim p(\boldsymbol{x}_t \mid \widehat{\boldsymbol{h}}_{t-1}, \widehat{\Theta})$ for $t = 2, \ldots, T$ and map back via $\widetilde{\boldsymbol{y}}_t = T^{-1}(\widetilde{\boldsymbol{x}}_t)$. We refer to this combined framework of LSTM and diffusion model as M_0 and use it for the final-phase submissions.

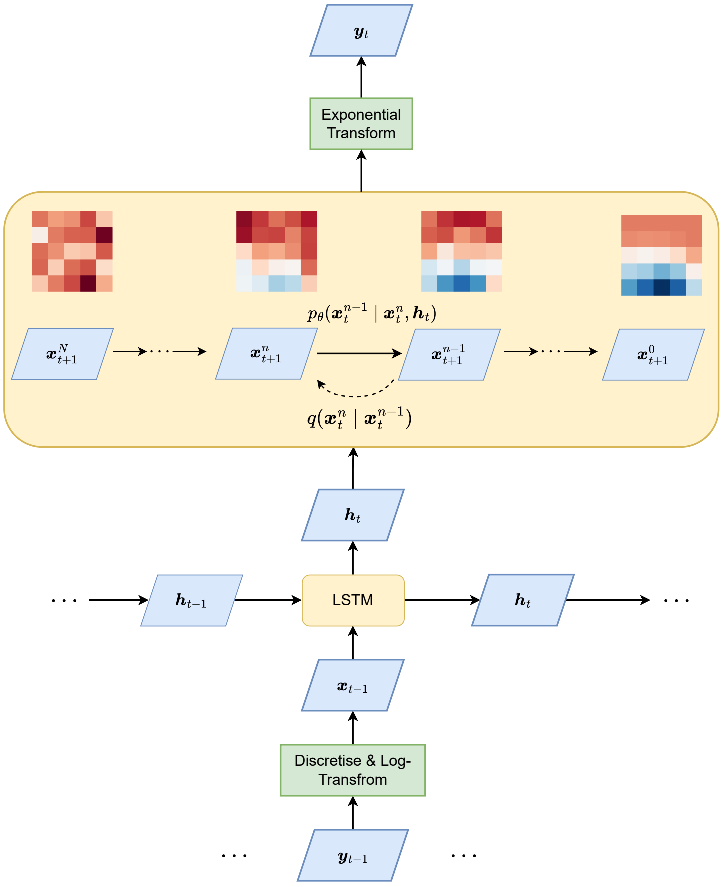

Figure 6.2: Architecture of the latent state framework for generating a new observation $\boldsymbol{y}_t$. Given the transformed input $\boldsymbol{x}_{t-1}$ and previous hidden state $\boldsymbol{h}_{t-1}$, the LSTM updates the hidden state to $\boldsymbol{h}_t$. Conditioning on $\boldsymbol{h}_t$, the diffusion model generates a transformed observation $\boldsymbol{x}_t$, which is then inverse-transformed to obtain $\boldsymbol{y}_t$.

6.3.5 Other explorations

We briefly describe the model used for the preliminary submission. Like M_0, it is a latent-state model with an LSTM encoder, but it replaces the conditional diffusion distribution with parametric, conditionally independent marginals. A similar idea was explored in Pasche and

Engelke (2024), where both a high quantile level in a quantile regression and parameters of a GPD for data exceeding that quantile are estimated using two LSTM models for tail extrapolation.

Model M_1 assumes that, conditional on the latent state $\boldsymbol{h}_{t-1}$, the 25 site-wise marginals $F_{s,t}$ are independent. To handle excess zeros we use a hurdle model:

$$F_{s,t}(y) = \begin{cases} 1 - \ p_0(s,t), & y = 0, \\ 1 - \ p_0(s,t) + p_0(s,t)\, G_{s,t}(y), & y > 0, \end{cases}$$

where $p_0(s,t) \in [0,1]$ is the probability of positive precipitation, modelled as a one-layer Multilayer perception (MLP) with sigmoid activation taking $\boldsymbol{h}_{t-1}$ as input.

For the positive part $G_{s,t}$, we splice a Weibull body and a GPD tail at the Weibull γ-quantile $q_{(s,t)}(\gamma)$:

$$G_{s,t}(y) = \begin{cases} F^{\text{Weibull}}_{s,t}\{y; \lambda(s,t), k(s,t)\}, & y < q_{(s,t)}(\gamma), \\ (1-\gamma)\, F^{\text{GPD}}_{s,t}\{y - q_{(s,t)}(\gamma); \sigma(s,t), \xi(s,t)\}, & y \geq q_{(s,t)}(\gamma), \end{cases}$$

with $\lambda, k, \sigma > 0$ and $\xi \in (-0.5, 0.5)$ (finite variance), all produced by a one-layer MLP from $\boldsymbol{h}_{t-1}$ using appropriate activations. We fix γ and tune it by grid search, since Bayesian analyses suggest the threshold posterior can be multimodal (Behrens et al., 2004; Hu et al., 2024), which can hinder gradient-based optimisation.

Under conditional independence, the log-likelihood factorises as

$$\sum_{t=2}^{T} \log f(\boldsymbol{y}_t; \boldsymbol{h}_{t-1}) = \sum_{t=2}^{T} \sum_{s=1}^{25} \log f_{s,t}(y_{s,t}; \boldsymbol{h}_{t-1}),$$

which we maximise over the LSTM/MLP parameters.

To sample from the fitted M1, we first compute $\widehat{p}_0(s,t)$ and the distributional parameters $\widehat{\lambda}(s,t)$, $\widehat{k}(s,t)$, $\widehat{\sigma}(s,t)$, $\widehat{\xi}(s,t)$ for all s and $t \geq 2$, then draw $U_1, U_2 \overset{\text{i.i.d.}}{\sim} \text{Unif}(0,1)$ and set

$$\widetilde{Y}_{s,t} = \begin{cases} 0, & U_1 > \widehat{p}_0(s,t), \\ \widehat{\lambda}(s,t)\{-\log(1-U_2)\}^{1/\widehat{k}(s,t)}, & U_1 \leq \widehat{p}_0(s,t),\ U_2 \leq \gamma, \\ \widehat{q}_{(s,t)}(\gamma) + \dfrac{\widehat{\sigma}(s,t)}{\widehat{\xi}(s,t)} \left[\left(1 - \dfrac{U_2 - \gamma}{1-\gamma}\right)^{-\widehat{\xi}(s,t)} - 1 \right], & U_1 \leq \widehat{p}_0(s,t),\ U_2 > \gamma. \end{cases}$$

6.4 Results

6.4.1 Evaluation metrics

The challenge targets six expected event counts $Q_1, \dots, Q_6$ (Section 6.2). Empirically, using the four available runs, $\widehat{Q}_i = 0$ for $i \neq 1$, so direct evaluation at the official thresholds is uninformative. Following Li et al. (2024), we instead view each target as a threshold-indexed functional $P_i(v)$ and assess models at lower v values where empirical counts are non-zero.

Let $\boldsymbol{Y} = (Y_{s,t}) \in \mathbb{R}^{25\times T}$ denote a random precipitation matrix with $T = 60{,}225$, row $\boldsymbol{Y}_t = (Y_{1,t}, \dots, Y_{25,t})$, and site sets $S_0 = \{1, \dots, 25\}$ (all sites) and $S_1 = \{1, \dots, 5\}$ (first grid row). For a threshold v, define

$$A_t(v) = \Big\{\boldsymbol{Y}_t : \textstyle\sum_{s=1}^{25} Y_{s,t} > v\Big\}, \qquad B_t^{(S,k)}(v) = \Big\{\boldsymbol{Y}_t : \#\{s \in S : Y_{s,t} > v\} \geq k\Big\}.$$

Then

$$\begin{aligned} P_1(v) &= \mathbb{E}\Big[\sum_{t=1}^{T} \mathbb{1}\{\boldsymbol{Y}_t \in A_t(v)\}\Big], \\ P_2(v) &= \mathbb{E}\Big[\sum_{t=1}^{T} \mathbb{1}\{\boldsymbol{Y}_t \in B_t^{(S_1,3)}(v)\}\Big], \\ P_4(v) &= \mathbb{E}\Big[\sum_{t=1}^{T} \mathbb{1}\{\min_{s\in S_0} Y_{s,t} > v\}\Big], \\ P_5(v) &= \mathbb{E}\Big[\sum_{t=1}^{T} \mathbb{1}\{\boldsymbol{Y}_t \in B_t^{(S_0,6)}(v)\}\Big]. \end{aligned}$$

The persistence-based quantities $P_3(v)$ and $P_6(v)$ count runs of length at least two consecutive days for the corresponding daily event indicators. Let

$$C_t^{(S,k)}(v) = \mathbb{1}\{\boldsymbol{Y}_t \in B_t^{(S,k)}(v)\}, \qquad t = 1, \dots, T,$$

and pad with $C_0^{(S,k)}(v) = C_{T+1}^{(S,k)}(v) = 0$. Define differences $D_t = C_t - C_{t-1}$ and the start/end indices of runs

$$\Phi = \{t : D_t = 1\}, \qquad \Psi = \{t : D_t = -1\},$$

with sorted elements $\Phi = \{\phi_1 < \cdots < \phi_m\}$ and $\Psi = \{\psi_1 < \cdots < \psi_m\}$. Run lengths are

$\ell_j = \psi_j - \phi_j$, so

$$P_3(v) = \mathbb{E}\Big[\sum_{j=1}^{m} \mathbb{1}\{\ell_j \geq 2\}\Big] \quad \text{with } (S, k) = (S_1, 3),$$
$$P_6(v) = \mathbb{E}\Big[\sum_{j=1}^{m} \mathbb{1}\{\ell_j \geq 2\}\Big] \quad \text{with } (S, k) = (S_0, 3).$$

The original targets are recovered by

$$Q_1 = P_1(85),\ Q_2 = P_2(4.3),\ Q_3 = P_3(2.5),\ Q_4 = P_4(1.7),\ Q_5 = P_5(5.7),\ Q_6 = P_6(5).$$

For evaluation we use 10 lower thresholds $\boldsymbol{v}_i$ for each P_i, chosen to yield non-zero empirical estimates:

$$\boldsymbol{v}_1 = (40.0, 43.3, 46.7, 50.0, 53.3, 56.7, 60.0, 63.3, 66.7, 70.0),$$
$$\boldsymbol{v}_2 = (2.0, 2.2, 2.4, 2.7, 2.9, 3.1, 3.3, 3.6, 3.8, 4.0),$$
$$\boldsymbol{v}_3 = (1.0, 1.1, 1.2, 1.3, 1.4, 1.6, 1.7, 1.8, 1.9, 2.0),$$
$$\boldsymbol{v}_4 = (0.80, 0.88, 0.97, 1.05, 1.13, 1.22, 1.30, 1.38, 1.47, 1.55),$$
$$\boldsymbol{v}_5 = (2.5, 2.7, 2.9, 3.2, 3.4, 3.6, 3.8, 4.1, 4.3, 4.5),$$
$$\boldsymbol{v}_6 = (1.8, 2.0, 2.2, 2.5, 2.7, 2.9, 3.1, 3.4, 3.6, 3.8).$$

Let $\widehat{P}_i(\boldsymbol{v}_i) \in \mathbb{R}^{10}$ be empirical estimates aggregated over runs 2–4, and let $\widetilde{P}_i(\boldsymbol{v}_i)$ be the corresponding estimates from simulations generated by a model trained on run1. We score fit by a range-scaled squared error,

$$R = \sum_{i=1}^{6} R_i, \qquad R_i = \frac{\|\widehat{P}_i(\boldsymbol{v}_i) - \widetilde{P}_i(\boldsymbol{v}_i)\|^2}{\max \widehat{P}_i(\boldsymbol{v}_i) - \min \widehat{P}_i(\boldsymbol{v}_i)}.$$

Lower R indicates better agreement across exceedance levels.

6.4.2 Model tuning and comparison

We first tune the preliminary model M_1 (Section 6.3.5), which uses an LSTM with a site-wise parametric likelihood. We proceed in two stages. (i) Fix the splice quantile γ in $q_{(s,t)}(\gamma)$ at $\gamma = 0.95$ and grid-search LSTM hyperparameters over number of layers $\{1, 2\}$ and hidden state

Table 6.1: Model comparison via the scaled squared error R_i between $\widetilde{P}_i(\boldsymbol{v}_i)$ and $\widehat{P}_i(\boldsymbol{v}_i)$; $R = \sum_{i=1}^{6} R_i$.

	R_1	R_2	R_3	R_4	R_5	R_6	R
M_0	**0.046**	**0.056**	**0.028**	**0.010**	**0.057**	0.055	**0.251**
M_1	0.064	0.059	0.029	0.069	0.062	**0.054**	0.336

size $\{64, 128, 256, 384\}$; the best score R is obtained with a two-layer LSTM and hidden size 384, which we keep thereafter. (ii) With this LSTM fixed, tune $\gamma \in \{0.60, 0.65, \ldots, 0.95\}$; the minimum R occurs at $\gamma = 0.80$, which defines the final M_1.

For M_0, we adopt the conditional diffusion architecture of Rasul et al. (2021a) for $\boldsymbol{\epsilon}_\theta$ in (6.2). In practice, R is far more sensitive to the LSTM than to the diffusion network, so for comparability we use the same two-layer/384 LSTM as in M_1. Since $\boldsymbol{h}_t$ is high-dimensional, we apply a linear projection to obtain a 25-dimensional conditioning vector for $\boldsymbol{\epsilon}_\theta$.

To compute each R_i, $\widehat{P}_i(\boldsymbol{v}_i)$ is estimated by averaging over runs 2–4, and $\widetilde{P}_i(\boldsymbol{v}_i)$ is estimated by the mean over 1,000 simulations from M_0 trained on run1. Table 6.1 reports $R_1, \ldots, R_6$ and $R = \sum_i R_i$. While differences for R_1–R_3 are small, M_0 achieves lower error on five of the six tasks. This is consistent with M_1's conditional-independence assumption across sites, which forces dependence to be expressed only through the shared latent state, whereas M_0 can represent residual multivariate dependence through the conditional diffusion component. Figure 6.3 shows $\widetilde{P}_i(\boldsymbol{v}_i)$ versus $\widehat{P}_i(\boldsymbol{v}_i)$; across tasks, curves align closely with no systematic bias.

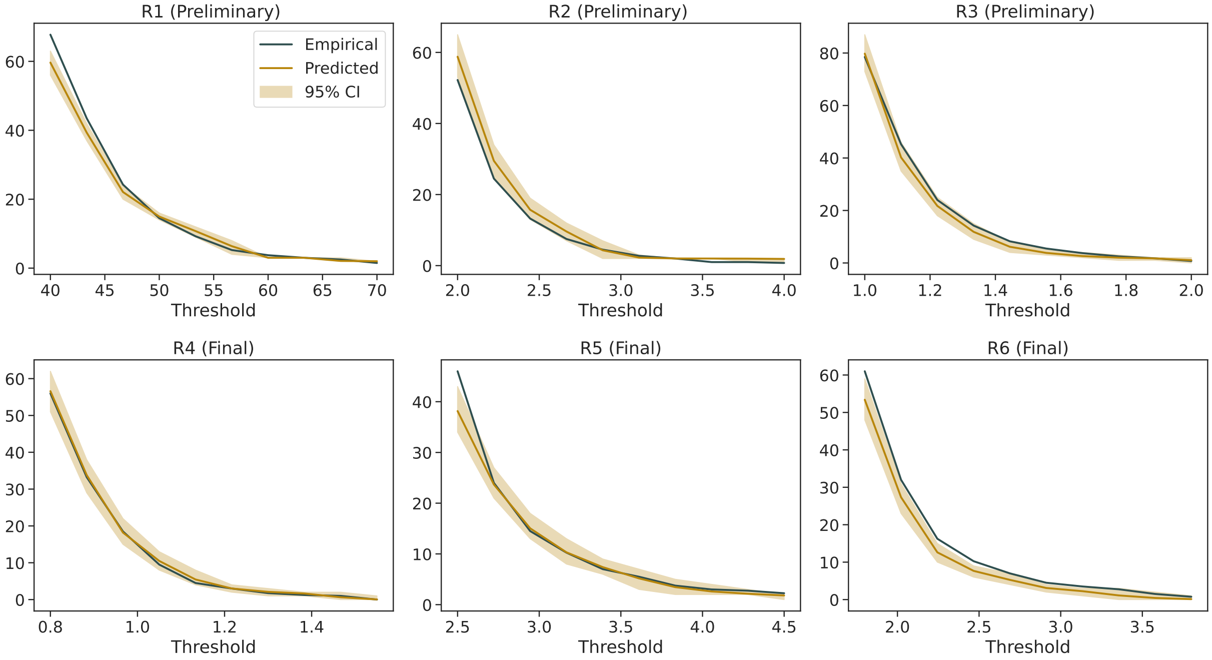
R1 (Preliminary)
R2 (Preliminary)
R3 (Preliminary)
R4 (Final)
R5 (Final)
R6 (Final)

Figure 6.3: Predicted $\widetilde{P}_i(\boldsymbol{v}_i)$ from M_0 (mean over 1,000 simulations; bands are 2.5%–97.5% simulation quantiles) versus empirical $\widehat{P}_i(\boldsymbol{v}_i)$ aggregated over runs 1–4, for $i = 1, \ldots, 6$.

To further check marginal tail behaviour, we estimate the GEV shape parameter ξ_{GEV} from annual maxima at each site using run1 and 1,000 simulations from M_0. Figure 6.4 shows close agreement at most sites; noticeable discrepancies are limited to sites 11 and 15 (with only marginal overlap of 95% intervals). Simulation-based intervals are systematically narrower than those from run1, plausibly because (i) the simulations condition on fitted parameters and do not propagate parameter uncertainty, and (ii) the expressive LSTM may yield overconfident latent states. We revisit these limitations in Section 6.5.

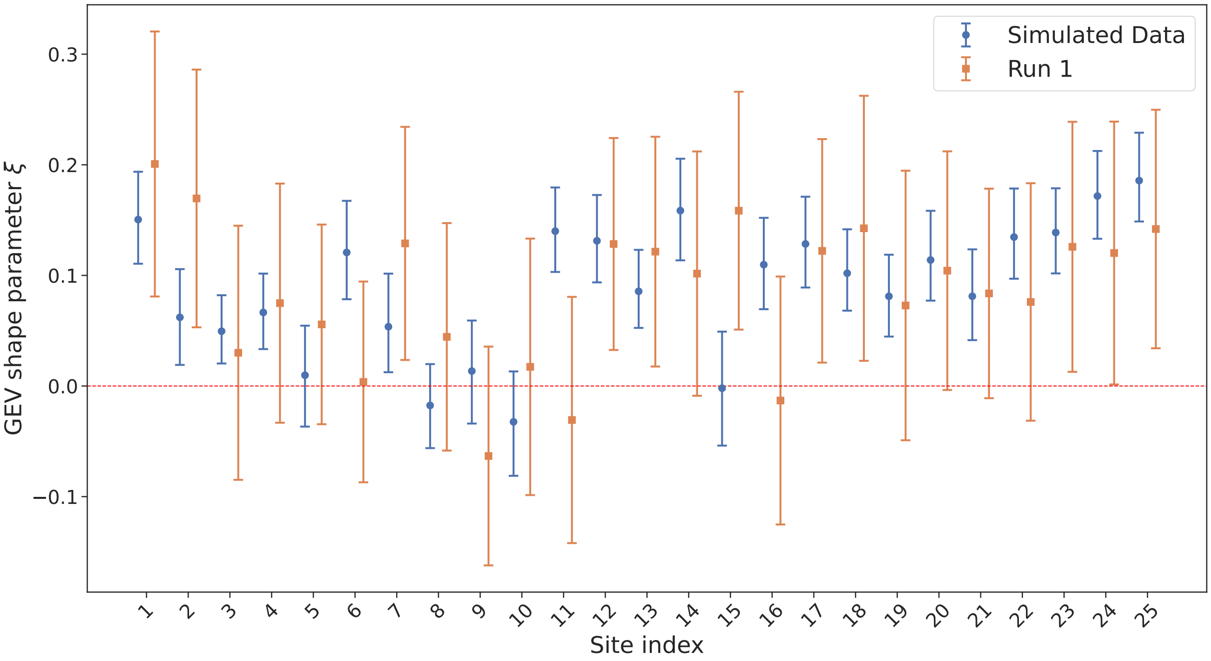

Figure 6.4: Site-wise ξ_{GEV} from annual maxima: run1 MLEs with asymptotic 95% CIs, and M_0 summaries from 1,000 simulated runs (mean with 2.5%–97.5% quantiles).

Finally, Figure 6.5 compares predicted and empirical marginal quantiles for each site (quantiles 0.5–0.999, excluding near-zero values). Agreement is good across sites, except for the highest quantiles at site 5. A recurring issue is that simulated quantile bands are unusually narrow, indicating underestimated predictive uncertainty, especially in the upper tail, which may degrade extrapolation and joint exceedance estimation (Section 6.5).

6.4.3 Final submission results

Final-phase quantities are computed from simulations of M_0. For uncertainty quantification, we use a temporal block bootstrap (no spatial resampling) to preserve within-block dependence. With $T = 60{,}225$ and block size b, we partition time into $K = \lceil T/b \rceil$ consecutive blocks (last possibly shorter); for each block index $k = 1, \ldots, K$, we draw (with replacement) one of the

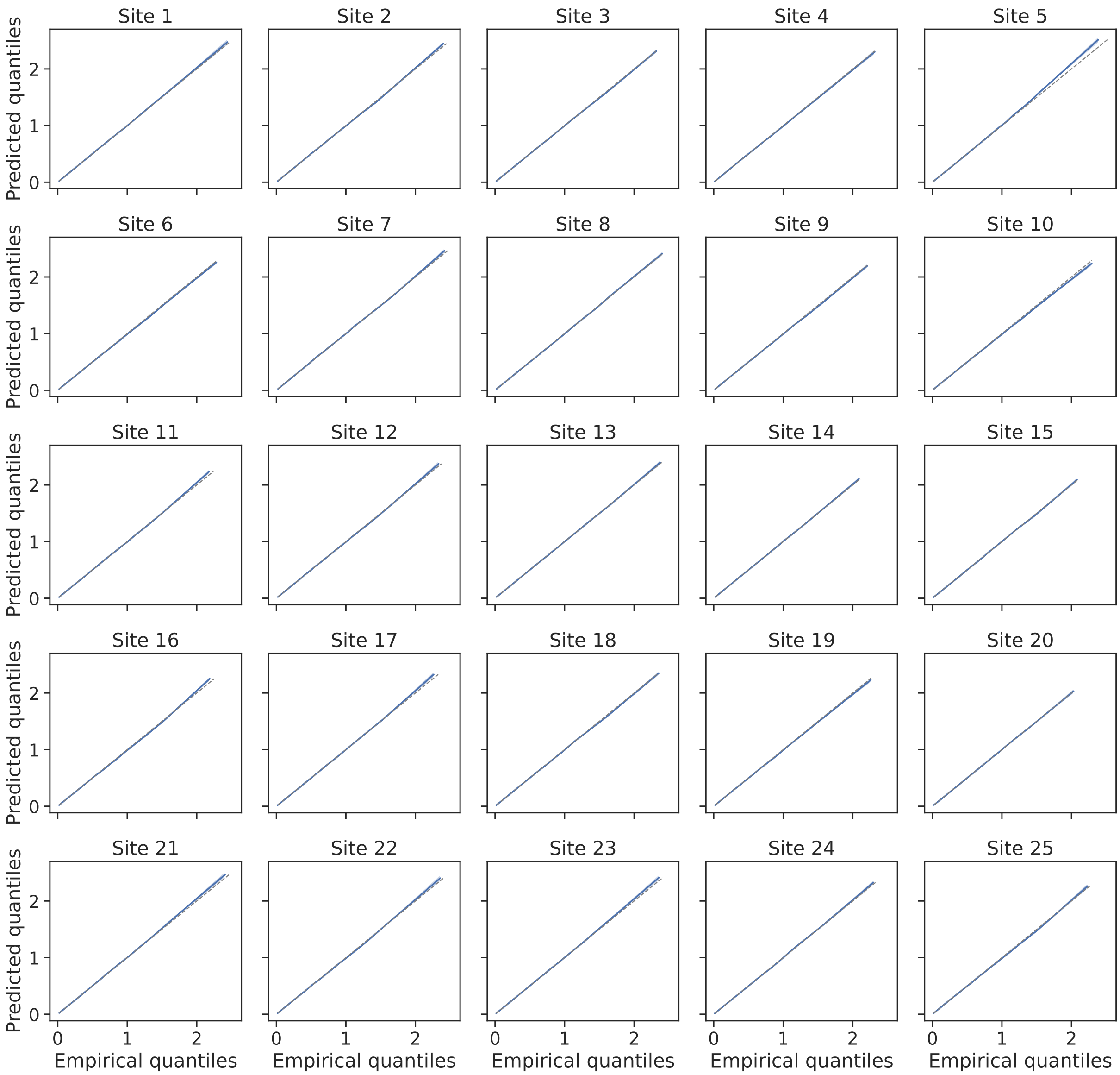

Figure 6.5: Site-wise Q–Q plots: M_0 quantiles versus empirical quantiles from run1; shaded regions are 95% simulation intervals from 1,000 simulated runs.

four runs' blocks at the same index k, and concatenate the selected blocks in order to form a bootstrap dataset. We choose b by grid search to minimise R (using $\widetilde{P}_i(\boldsymbol{v}_i)$ on bootstrap samples), obtaining $b = 8000$. We generate 400 bootstrap datasets; for each, we fit M_0 and simulate 1,000 runs to estimate Q_4, Q_5, Q_6. Point estimates are bootstrap means and 95% CIs are bootstrap 2.5%–97.5% quantiles (Table 6.2).

Table 6.2: Final submission: 400 temporal block bootstrap samples ($b = 8000$). Point estimates are bootstrap means; 95% CIs are bootstrap 2.5%–97.5% quantiles.

Quantity	Point Estimate	95% Confidence Interval
Q_4	0.02	(0.00, 0.08)
Q_5	0.12	(0.02, 0.68)
Q_6	0.32	(0.02, 1.42)

6.5 Conclusion and discussion

We proposed a deep generative approach for modelling daily precipitation on a 25-site grid over $T = 60{,}225$ days, with emphasis on joint high-threshold exceedances. Temporal dependence is encoded by an LSTM, and conditional on the latent state we compare: M_0, a diffusion model applied to jittered-log transformed data; and M_1, a conditionally independent baseline with hurdle marginals and a Weibull-GPD splice. Using six exceedance-based functionals evaluated at lower thresholds, M_0 outperforms M_1 on five metrics and reproduces marginal tail shape reasonably well, as assessed by GEV shape estimates from annual maxima. Final submissions were obtained by fitting M_0 across 400 temporal block bootstrap samples.

6.5.1 Discussion

Diffusion models. Diffusion models play two roles in our framework. First, they provide a flexible, continuous conditional likelihood for positive-valued precipitation, so the bulk and tail are modelled within a single smooth distributional family on the transformed scale rather than by piecing together two distributions with an explicit threshold splice. This contrasts with GPD-based constructions, where threshold splicing induces discontinuities in the marginal models at a chosen threshold. Within latent-variable extreme-value frameworks, an alternative route to continuity is to adopt a continuous marginal family that has asymptotic tail properties from the EVT (e.g., extended generalised Pareto forms), as explored by Cisneros et al. (2024).

Second, diffusion models can represent high-dimensional spatial dependence without imposing a fixed parametric conditional dependence structure. Instead, dependence is learned jointly with the marginal distributions through the backward propagation, which can be advantageous when the dependence pattern varies across regimes. A caveat is that, absent tail-specific objectives or constraints, diffusion models trained by standard denoising losses may underrepresent very rare extremes. In our setting, the jittered-log transformation and the LSTM-conditioned latent-state formulation can empirically stabilise training and improve tail behaviour, but they do not guarantee correct far-tail extrapolation; this motivates the diagnostic focus on exceedance-based metrics and annual-maxima shape estimates reported above.

Role of the latent space. In both M_0 and M_1, the observation model is conditioned on the LSTM hidden state. This latent state captures temporal dependence and regime information and can encode predictors of conditional magnitude (e.g., time-varying location), thereby providing a strong "context" for the conditional likelihood. When the LSTM state is highly expressive, much of the spatiotemporal structure is absorbed into the latent representation, and the conditional likelihood primarily learns residual variation around that context. Consistent with this interpretation, we found that increasing the LSTM state dimension/depth had a larger impact on predictive performance than tuning the likelihood-model hyperparameters, and that M_0 and M_1 perform similarly on several metrics when the LSTM configuration is held fixed (Table 6.1). A similar phenomenon is also found in Chapter 5 under a Bayesian latent Gaussian framework for wildfire modelling.

6.5.2 Limitation

Two limitations are most salient. (i) Extrapolation instability: at very high quantiles, estimates of Q_i can vary substantially across random seeds when fitting M_0 on run1 with fixed hyperparameters, caused by the simulation-based estimation and data sparsity in the far tail. Variance reduction via ensembling/bootstrapping is therefore needed but computationally costly. (ii) Underestimated uncertainty: simulation-based bands for quantities such as ξ_{GEV} and marginal quantiles are often overly narrow (Section 6.4.2), likely because parameter uncertainty is ignored and because an expressive LSTM may produce near-deterministic latent states, leaving the conditional diffusion model to learn only small residual noise. A simple train/validation split by runs was not effective: models captured mean structure but missed tail behaviour. A possible alternative is to condition the diffusion model on lower-capacity, shared temporal covariates (e.g., day-of-year or week-of-year) rather than the full LSTM state, which may better balance regularisation and tail fit.

Chapter 7

Conclusion & future work

In this thesis, starting from the idea of generalising the univariate piecing-together model to the multivariate setting, we develop three distinct frameworks to achieve the goal of jointly describing the multivariate bulk and tail data, together with one framework designed to address the infinite parameterisation problem in multivariate threshold exceedance modelling.

Chapter 3 presents a direct generalisation of the univariate joint modelling approach that combines a parametric or semi-parametric distribution for the bulk and a GPD for the tail. We demonstrate our work by combining a bivariate Gaussian and an mGPD with an independent reverse exponential generator. A criticism of this framework is that the bivariate Gaussian exhibits asymptotic independence (AI) while the mGPD exhibits asymptotic dependence (AD), which seems to make the use of the mGPD for tail approximation questionable. However, this design is intentional: it ensures robust tail characterisation for small datasets. Through simulations, we show that the model performs reasonably well even when the data are AI. We also discuss alternative bulk distributions, such as constructing a multivariate distribution via customised copula–marginal combinations. One might consider using a vine copula (Kurowicka and Joe, 2010) or normalising flows to achieve a more flexible dependence structure. Unfortunately, these methods cannot be incorporated into the bivariate extreme mixture model framework, as they lack an explicit form for the CDF values, which is essential for determining threshold exceedance probabilities. For the tail, other parametric forms of the mGPD are possible, but constrained by their limiting forms; they still lack expressiveness in modelling tail dependence. This limitation directly motivates our work in Chapter 4.

Overall, the bivariate extreme mixture model represents a first attempt at multivariate joint modelling of bulk and tail. It shows good performance in estimating tail shape and bulk–tail dependence through simulations and an application to UK air temperature data at two nearby sites. Nevertheless, it has drawbacks, such as the abrupt cutoff of the joint density at the threshold and the restriction that the tail is always AD. For this reason, after completing Chapter 4, we move beyond this framework to explore alternative approaches for joint bulk–tail modelling.

Chapter 4 introduces GPDFlow, a framework that parameterises the generator T in the mGPD using normalising flows, thereby providing a learnable and flexible dependence structure. Unlike classical mGPD models, GPDFlow requires no explicit specification of T, and the learnt T exhibits far richer forms than those previously studied. Its flexible dependence also enables inference via the full likelihood, rather than the censored likelihood typically used to reduce bias in dependence estimation from points near the lower endpoint of the mGPD. This feature makes GPDFlow suitable for computing partial exceedance probabilities, provided diagnostic checks support its application.

GPDFlow also contributes to the field of generative models: it demonstrates that a single transformation can capture marginal tail heaviness if the remaining transformations map the base distribution into a standard form (e.g. standard mGPD). This design enhances interpretability, with tail heaviness clearly represented by the marginal tail shape parameters. Extensive simulation studies show that GPDFlow yields accurate estimates of marginal tail heaviness, tail dependence, and partial exceedance probabilities. In an empirical application, GPDFlow outperforms the classical mGPD in modelling margins and dependence of exceedance data from the negative log-returns of five US banks.

One limitation, however, is that GPDFlow's flexible tail dependence remains restricted to the AD setting. By construction, due to the transformation derived from the stochastic representation of the mGPD, GPDFlow is always AD, irrespective of the normalising flow structure. This restriction makes it unsuitable for high-dimensional problems, where AD assumptions rarely hold. The subsequent chapters, therefore, focus on addressing this limitation.

In Chapter 5, we turn to a high-dimensional spatial application: forecasting fire counts, and moderate and extreme burnt areas, one month ahead across 278 Portuguese counties. To overcome rigid tail dependence in multivariate EVT, we adopt a latent Gaussian model to implicitly capture dependence across observations (fire counts and burnt areas). Moderate and extreme burnt areas are jointly modelled via the sub-asymptotic eGP likelihood, which also resolves the discontinuity issue since the eGP density requires no threshold. We employ INLA for efficient Bayesian inference in this latent Gaussian model and carefully discuss PC priors for the eGP hyperparameters.

This approach achieves half of our joint bulk–tail modelling goal. For forecasting, we want to incorporate environmental covariates (e.g. FWI, air temperature), which are strongly associated with wildfires, to improve the forecast accuracy. However, the latent Gaussian model cannot capture complex covariate interactions, and robust INLA implementation requires limiting the number of covariates to avoid excessive hyperparameters. Additionally, obtaining environmental covariates one month ahead is generally challenging. We therefore use an XGBoost model to capture patterns among environmental covariates and wildfire histories, producing one-month-ahead council-level predictions of fire counts and burnt areas. These are treated as pseudo-covariates in the latent Gaussian model with INLA. This hybrid approach leverages the strengths of both models: XGBoost effectively captures complex dependencies but lacks uncertainty quantification, while the latent Gaussian model naturally provides uncertainty quantification via Bayesian inference

When comparing eGP likelihood against alternative Gamma and Weibull likelihoods, only marginal improvements are observed. We attribute this to the model structure: with a sufficiently expressive linear predictor, extreme values are explained primarily by the predictor itself. This makes us reflect on the necessity of EVT in such contexts.

Finally, in Chapter 6, we explore a deep learning approach independent of EVT to model daily precipitation over a 5×5 grid for 165 years of climate-model simulations, and compare its performance with an EVT-based alternative. We use an LSTM to encode spatio-temporal information into hidden vectors and condition on these to fit:

1. a conditional diffusion model applied to logarithm-transformed data with zero adjustment, characterising dependence across grid cells; and

2. a conditional independent distribution with sliced margins comprising a Weibull distribution for the bulk and a GPD for the tail.

The diffusion model-based approach achieves superior performance in five out of six metrics derived from the expected number of extreme events. We also compare marginal tail shapes of simulated data from the conditional diffusion model with the true one, by fitting a GEV to annual block maxima and examining shape parameter estimates. Results show that the deep learning approach accurately estimates tail shape, with estimated parameters aligning closely with those from the true data. However, the 95% confidence intervals of the GEV shape parameters from simulated data are narrower than those from the actual data, which is an inherent limitation of deep learning models due to their lack of parameter uncertainty.

7.1 Future work

In the past five years, the number of studies applying deep learning techniques to extreme value analysis has grown rapidly. A key reason is the success of deep learning in fields such as computer vision and natural language processing, making it natural to combine deep learning with other disciplines. We regard deep learning as a great opportunity to bring new insights to the statistical study of extremes, hence the two chapters in this thesis devoted to this direction. Our results show that such a combination can address longstanding challenges in EVT, such as complex tail dependence.

Future research at the intersection of EVT and deep learning can proceed in two main directions. First, EVT can serve as the foundational framework, with deep learning acting as a modelling component or inference tool to address challenges that are difficult to resolve in classical EVT. Tail dependence provides a clear example: while its structure is complex, it does not involve aspects that are inherently hard for deep learning to estimate, such as heavy tails, making it particularly suitable for approximation by deep learning models. A current focus in modern multivariate EVT is the radius–angular representation of extremes, where observations are transformed into polar coordinates so that extremes are captured solely in the radial component, facilitating the modelling of extremes in any direction. Several studies have already explored the use of deep learning for modelling tail dependence in the radius–angular representation (Murphy-Barltrop et al., 2024; De Monte et al., 2025; Mackay et al., 2024).

Deep learning is also emerging as a powerful inference tool in EVT. Max-stable processes, such as the Brown–Resnick model, generalise multivariate EVT to infinite dimensions but face limited application in spatial problems due to their intractable density functions and computational demands. Deep learning models can act as surrogate models, directly mapping observations to parameters and enabling likelihood-free inference. Related works include Lenzi et al. (2023) and Richards et al. (2024).

Second, deep learning can serve as the base framework, with EVT mitigating its known drawbacks, particularly its difficulty in handling heavy-tailed data. In the field of generative models, numerous studies have investigated the learning and generation of heavy-tailed data (Allouche et al., 2022; McDonald et al., 2022), but explicit and stable estimation of the tail index remains an open challenge. This has been partly addressed in our GPDFlow framework and in Hickling and Prangle (2024), where tail heaviness is entirely captured by the outermost layer of normalising flows, specifically designed to transform light-tailed distributions into heavy-tailed ones. However, in GPDFlow (though not discussed in the main chapter), we observe that bias in estimated shape parameters increases with dimension, which further affects dependence estimation. For example, in an experiment with $d = 25$, GPDFlow was unable to reliably estimate the tail coefficient χ when marginal shape and scale parameters were free, whereas the issue did not arise when marginal parameters were fixed. This suggests that single-layer transformations are insufficient for accurately estimating the tail index in high-dimensional settings. Future work could explore gradual transformations of the margins to alter tail heaviness progressively as the layer of the deep learning model goes deeper, rather than in a one-step manner.

Another promising direction is to investigate whether censored likelihood inference can enhance tail estimation in generative models. Existing frameworks for modelling heavy-tailed data with generative models have focused primarily on architectural design to capture samples at high quantiles, with less attention given to modifying the objective function. In EVT, censored likelihood inference is a common technique to reduce the influence of bulk data on tail estimation (Huser et al., 2016). Adapting this censoring approach to likelihood-based generative models (e.g. normalising flows) could reveal whether censoring improves performance when the primary interest lies in the tail region.

Bibliography

M. B. Alaya, F. Zwiers, X. Zhang, An evaluation of block-maximum-based estimation of very long return period precipitation extremes with a large ensemble climate simulation, Journal of Climate 33 (2020) 6957–6970.

S. Allen, D. Ginsbourger, J. Ziegel, Evaluating forecasts for high-impact events using transformed kernel scores, SIAM/ASA Journal on Uncertainty Quantification 11 (2023) 906–940.

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

L. André, J. Wadsworth, A. O'Hagan, Joint modelling of the body and tail of bivariate data, Computational Statistics & Data Analysis 189 (2024) 107841.

S. Aulbach, V. Bayer, M. Falk, A multivariate piecing-together approach with an application to operational loss data, Bernoulli 18 (2012a).

S. Aulbach, M. Falk, M. Hofmann, The multivariate piecing-together approach revisited, Journal of Multivariate Analysis 110 (2012b) 161–170.

A. A. Balkema, L. De Haan, Residual life time at great age, The Annals of probability (1974) 792–804.

J. Barnard, R. McCulloch, X.-L. Meng, Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage, Statistica Sinica 10 (2000) 1281–1311.

C. N. Behrens, H. F. Lopes, D. Gamerman, Bayesian analysis of extreme events with threshold estimation, Statistical modelling 4 (2004) 227–244.

J. Beirlant (Ed.), Statistics of extremes, Wiley series in probability and statistics, Wiley, Hoboken, NJ, 2004.

J. Beirlant, Y. Goegebeur, J. Segers, J. L. Teugels, Statistics of extremes: theory and applications, John Wiley & Sons, 2006.

P. Billingsley, Probability and measure, A Wiley-Interscience publication, Wiley, New York [u.a.], 3. ed edition, 1995.

C. M. Bishop, N. M. Nasrabadi, Pattern recognition and machine learning, volume 4, Springer, 2006.

P. Bortot, S. Coles, J. Tawn, The multivariate gaussian tail model: An application to oceanographic data, Journal of the Royal Statistical Society: Series C (Applied Statistics) 49 (2000) 31–049.

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

L. Breiman, Random forests, Machine learning 45 (2001) 5–32.

M. Castellanos, S. Cabras, A default Bayesian procedure for the generalized Pareto distribution, Journal of Statistical Planning and Inference 137 (2007) 473–483.

D. Castro-Camilo, R. Huser, H. Rue, A spliced gamma-generalized pareto model for short-term extreme wind speed probabilistic forecasting, Journal of Agricultural, Biological and Environmental Statistics 24 (2019) 517–534.

V. Chavez-Demoulin, A. C. Davison, Generalized additive modelling of sample extremes, Journal of the Royal Statistical Society Series C: Applied Statistics 54 (2005) 207–222.

T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, p. 785–794.

D. Cisneros, Y. Gong, R. Yadav, A. Hazra, R. Huser, A combined statistical and machine learning approach for spatial prediction of extreme wildfire frequencies and sizes, Extremes 26 (2023) 301–330.

D. Cisneros, J. Richards, A. Dahal, L. Lombardo, R. Huser, Deep graphical regression for jointly moderate and extreme Australian wildfires, Spatial Statistics 59 (2024) 100811.

S. Coles, J. Bawa, L. Trenner, P. Dorazio, An introduction to statistical modeling of extreme values, volume 208, Springer, 2001.

S. Coles, J. Heffernan, J. Tawn, Dependence measures for extreme value analyses, Extremes 2 (1999) 339–365.

S. G. Coles, E. A. Powell, Bayesian Methods in Extreme Value Modelling: A Review and New Developments, International Statistical Review / Revue Internationale de Statistique 64 (1996) 119–136.

C. C. DaCamara, The signature of climate in annual burned area in portugal, Climate 12 (2024) 143.

C. C. DaCamara, T. J. Calado, S. L. Ermida, I. F. Trigo, M. Amraoui, K. F. Turkman, Calibration of the fire weather index over mediterranean europe based on fire activity retrieved from msg satellite imagery, International Journal of Wildland Fire 23 (2014) 945–958.

C. C. DaCamara, R. M. Trigo, M. M. Pinto, S. A. Nunes, I. F. Trigo, C. M. Gouveia, M. Rainha, Ceasefire: a website to assist fire managers in portugal, Advances in Forest Fire Research 2108 (2018) 941–949.

A. C. Davison, R. L. Smith, Models for Exceedances Over High Thresholds, Journal of the Royal Statistical Society: Series B (Methodological) 52 (1990) 393–425.

L. De Haan, A. Ferreira, Extreme value theory: an introduction, Springer, 2006.

L. De Monte, R. Huser, I. Papastathopoulos, J. Richards, Generative modelling of multivariate geometric extremes using normalising flows, arXiv preprint arXiv:2505.02957 (2025).

P. de Valpine, D. Turek, C. Paciorek, C. Anderson-Bergman, D. Temple Lang, R. Bodik, Programming with models: writing statistical algorithms for general model structures with NIMBLE, Journal of Computational and Graphical Statistics 26 (2017) 403–413. R package version 4.3.1.

L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using real NVP, in: International Conference on Learning Representations.

F. F. do Nascimento, D. Gamerman, H. F. Lopes, A semiparametric Bayesian approach to extreme value estimation, Statistics and computing 22 (2012) 661–675.

Duvsten Östin, Hanna and Gasslander, Tilda, Predicting wildfires: Spatio-temporal Modeling of Wildfires in Maule, Chile, and Analysis of the Risk Communication Tool Botón Rojo, 2025. Student Paper.

S. Elsayed, D. Thyssens, A. Rashed, H. S. Jomaa, L. Schmidt-Thieme, Do we really need deep learning models for time series forecasting?, arXiv preprint arXiv:2101.02118 (2021).

M. Falk, G. Stupfler, An offspring of multivariate extreme value theory: The max-characteristic function, Journal of Multivariate Analysis 154 (2017) 85–95.

R. A. Fisher, L. H. C. Tippett, Limiting forms of the frequency distribution of the largest or smallest member of a sample, Mathematical Proceedings of the Cambridge Philosophical Society 24 (1928) 180–190.

J. H. Friedman, Greedy function approximation: a gradient boosting machine, Annals of statistics (2001) 1189–1232.

A. Frigessi, O. Haug, H. Rue, A dynamic mixture model for unsupervised tail estimation without threshold selection, Extremes 5 (2002) 219–235.

E. Gabriel, T. Opitz, F. Bonneu, Detecting and modeling multi-scale space-time structures: the case of wildfire occurrences, Journal de la Société Française de Statistique 158 (2017) 86–105.

A. E. Gelfand, A. F. Smith, Sampling-based approaches to calculating marginal densities, Journal of the American statistical association 85 (1990) 398–409.

A. Gelman, D. B. Rubin, Inference from iterative simulation using multiple sequences, Statistical Science 7 (1992) 457–472.

S. Geman, D. Geman, Stochastic relaxation, gibbs distributions, and the bayesian restoration of images, IEEE Transactions on pattern analysis and machine intelligence (1984) 721–741.

T. Gneiting, A. E. Raftery, Strictly proper scoring rules, prediction, and estimation, Journal of the American Statistical Association 102 (2007) 359–378.

Y. Gorishniy, I. Rubachev, V. Khrulkov, A. Babenko, Revisiting deep learning models for tabular data, Advances in neural information processing systems 34 (2021) 18932–18943.

Government of Portugal, Portugal's Adaptation Communication to the United Nations Framework Convention on Climate Change, Technical Report, Government of Portugal, 2021.

L. Grinsztajn, E. Oyallon, G. Varoquaux, Why do tree-based models still outperform deep learning on typical tabular data?, Advances in neural information processing systems 35 (2022) 507–520.

L. Haan, A. Ferreira, Extreme value theory: an introduction, volume 3, Springer, 2006.

W. K. Hastings, Monte carlo sampling methods using markov chains and their applications, Biometrika 57 (1970) 97–109.

Y. He, L. Peng, D. Zhang, Z. Zhao, Risk analysis via generalized pareto distributions, Journal of Business & Economic Statistics 40 (2022) 852–867.

J. E. Heffernan, J. A. Tawn, A conditional approach for multivariate extreme values (with discussion), Journal of the Royal Statistical Society Series B: Statistical Methodology 66 (2004) 497–546.

T. Hickling, D. Prangle, Flexible tails for normalizing flows, arXiv preprint arXiv:2406.16971 (2024).

J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Advances in neural information processing systems 33 (2020) 6840–6851.

S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (1997) 1735–1780.

K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural networks 2 (1989) 359–366.

C. Hu, D. Castro-Camilo, Gpdflow: Generative multivariate threshold exceedance modeling via normalizing flows, arXiv preprint arXiv:2503.11822 (2025).

C. Hu, B. Swallow, D. Castro-Camilo, A Bayesian multivariate extreme value mixture model, arXiv preprint arXiv:2401.15703 (2024).

W. K. Huang, D. W. Nychka, H. Zhang, Estimating precipitation extremes using the log-histospline, Environmetrics 30 (2019) e2543.

R. Huser, A. C. Davison, M. G. Genton, Likelihood estimators for multivariate extremes, Extremes 19 (2016) 79–103.

R. Huser, T. Opitz, J. L. Wadsworth, Modeling of spatial extremes in environmental data science: Time to move away from max-stable processes, Environmental Data Science 4 (2025) e3.

R. Huser, J. L. Wadsworth, Modeling spatial processes with unknown extremal dependence class, Journal of the American statistical association 114 (2019) 434–444.

S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International conference on machine learning, pmlr, pp. 448–456.

R. M. Iverson, Landslide triggering by rain infiltration, Water resources research 36 (2000) 1897–1910.

P. Jaini, I. Kobyzev, Y. Yu, M. A. Brubaker, Tails of lipschitz triangular flows, in: Proceedings of the 37th International Conference on Machine Learning, ICML'20, JMLR.org, 2020.

T. Januschowski, Y. Wang, K. Torkkola, T. Erkkilä, H. Hasson, J. Gasthaus, Forecasting with trees, International Journal of Forecasting 38 (2022) 1473–1481.

B. Jørgensen, Exponential dispersion models, Journal of the Royal Statistical Society Series B: Statistical Methodology 49 (1987) 127–145.

D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations (ICLR).

D. P. Kingma, P. Dhariwal, Glow: Generative flow with invertible 1x1 convolutions, Advances in neural information processing systems 31 (2018).

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

J. Koh, Gradient boosting with extreme-value theory for wildfire prediction, Extremes 26 (2023) 273–299.

J. Koh, F. Pimont, J.-L. Dupuy, T. Opitz, Spatiotemporal wildfire modeling through point processes with moderate and extreme marks, The annals of applied statistics 17 (2023) 560–582.

E. Krainski, V. Gómez-Rubio, H. Bakka, A. Lenzi, D. Castro-Camilo, D. Simpson, F. Lindgren, H. Rue, Advanced spatial modeling with stochastic partial differential equations using R and INLA, Chapman and Hall/CRC, 2018.

M. Krock, J. Bessac, M. L. Stein, A. H. Monahan, Nonstationary seasonal model for daily mean temperature distribution bridging bulk and tails, Weather and Climate Extremes 36 (2022) 100438.

S. Kullback, R. A. Leibler, On information and sufficiency, The annals of mathematical statistics 22 (1951) 79–86.

D. Kurowicka, H. Joe, Dependence modeling: vine copula handbook, World Scientific, 2010.

M. J. Van der Laan, E. C. Polley, A. E. Hubbard, Super learner, Statistical applications in genetics and molecular biology 6 (2007).

E. S. Lawler, B. A. Shaby, Anthropogenic and meteorological effects on the counts and sizes of moderate and extreme wildfires, Environmetrics 35 (2024) e2873.

M. R. Leadbetter, Extremes and local dependence in stationary sequences, Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete 65 (1983) 291–306.

A. Lenzi, J. Bessac, J. Rudi, M. L. Stein, Neural networks for parameter estimation in intractable models, Computational Statistics & Data Analysis 185 (2023) 107762.

M. Leonelli, D. Gamerman, Semiparametric bivariate modelling with flexible extremal dependence, Statistics and computing 30 (2020) 221–236.

D. Lewandowski, D. Kurowicka, H. Joe, Generating random correlation matrices based on vines and extended onion method, Journal of Multivariate Analysis 100 (2009) 1989–2001.

S. Lhaut, H. Rootzén, J. Segers, Wasserstein-aitchison gan for angular measures of multivariate extremes, arXiv preprint arXiv:2504.21438 (2025).

M. Li, D. Cuba, C. Hu, D. Castro-Camilo, A wee exploration of techniques for risk assessments of extreme events: EVA (2023) conference data challenge: wee extremes group, Extremes (2024) 1–21.

B. Lim, S. Ö. Arık, N. Loeff, T. Pfister, Temporal fusion transformers for interpretable multihorizon time series forecasting, International journal of forecasting 37 (2021) 1748–1764.

S. M. Lundberg, G. G. Erion, S.-I. Lee, Consistent individualized feature attribution for tree ensembles, arXiv preprint arXiv:1802.03888 (2018).

S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, p. 4768–4777.

C. MacBride, V. Davies, D. Lee, A spatial autoregressive random forest algorithm for small-area spatial prediction, The Annals of Applied Statistics 19 (2025) 485–504.

A. MacDonald, C. J. Scarrott, D. Lee, B. Darlow, M. Reale, G. Russell, A flexible extreme value mixture model, Computational statistics & data analysis 55 (2011) 2137–2157.

E. Mackay, C. Murphy-Barltrop, P. Jonathan, The SPAR model: A new paradigm for multivariate extremes: Application to joint distributions of metocean variables, Journal of Offshore Mechanics and Arctic Engineering 147 (2025) 011205.

E. Mackay, C. Murphy-Barltrop, J. Richards, P. Jonathan, Deep learning joint extremes of metocean variables using the SPAR model, arXiv preprint arXiv:2412.15808 (2024).

G. Mainik, E. Schaanning, On dependence consistency of covarand some other systemic risk measures, Statistics & Risk Modeling 31 (2014) 49–77.

A. McDonald, P.-N. Tan, L. Luo, Comet flows: Towards generative modeling of multivariate extremes and tail dependence, arXiv preprint arXiv:2205.01224 (2022).

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, Equation of state calculations by fast computing machines, The journal of chemical physics 21 (1953) 1087–1092.

C. Murphy, J. A. Tawn, Z. Varty, Automated threshold selection and associated inference uncertainty for univariate extremes, Technometrics (2024) 1–10.

C. J. Murphy-Barltrop, R. Majumder, J. Richards, Deep learning of multivariate extremes via a geometric representation, arXiv preprint arXiv:2406.19936 (2024).

P. Naveau, R. Huser, P. Ribereau, A. Hannart, Modeling jointly low, moderate, and heavy rainfall intensities without a threshold selection, Water Resources Research 52 (2016) 2753–2769.

P. Naveau, J. Segers, Multivariate extreme value theory, arXiv preprint arXiv:2412.18477 (2024).

R. M. Neal, Slice sampling, The annals of statistics 31 (2003) 705–767.

J. van Niekerk, H. Rue, Low-rank variational bayes correction to the laplace method, Journal of Machine Learning Research 25 (2024) 1–25.

S. A. Nunes, C. C. DaCamara, J. M. Pereira, R. M. Trigo, Assessing the role played by meteorological conditions on the interannual variability of fire activity in four subregions of iberia, International journal of wildland fire 32 (2023) 1529–1541.

T. Opitz, Latent gaussian modeling and inla: A review with focus on space-time applications, Journal de la société française de statistique 158 (2017) 62–85.

T. Opitz, Eva 2021 data challenge on spatiotemporal prediction of wildfire extremes in the usa, Extremes 26 (2023) 241–250.

T. Opitz, F. Bonneu, E. Gabriel, Point-process based bayesian modeling of space–time structures of forest fire occurrences in mediterranean france, Spatial Statistics 40 (2020) 100429.

T. Opitz, R. Huser, H. Bakka, H. Rue, INLA goes extreme: Bayesian tail regression for the estimation of high spatio-temporal quantiles, Extremes 21 (2018a) 441–462.

T. Opitz, R. Huser, H. Bakka, H. Rue, Inla goes extreme: Bayesian tail regression for the estimation of high spatio-temporal quantiles, Extremes 21 (2018b) 441–462.

K. Pandey, J. Pathak, Y. Xu, S. Mandt, M. Pritchard, A. Vahdat, M. Mardani, Heavy-tailed diffusion models, arXiv preprint arXiv:2410.14171 (2024).

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

G. Papamakarios, T. Pavlakou, I. Murray, Masked autoregressive flow for density estimation, Advances in neural information processing systems 30 (2017).

I. Papastathopoulos, J. A. Tawn, Extended generalised pareto models for tail estimation, Journal of Statistical Planning and Inference 143 (2013) 131–143.

O. C. Pasche, S. Engelke, Neural networks for extreme quantile regression with an application to forecasting of flood risk, The Annals of Applied Statistics 18 (2024) 2818–2839.

F. Pimont, H. Fargeon, T. Opitz, J. Ruffault, R. Barbero, N. Martin-StPaul, E. Rigolot, M. Riviere, J.-L. Dupuy, Prediction of regional wildfire activity in the probabilistic bayesian framework of firelihood, Ecological applications 31 (2021) e02316.

M. M. Pinto, C. C. DaCamara, I. F. Trigo, R. M. Trigo, K. F. Turkman, Fire danger rating over mediterranean europe based on fire radiative power derived from meteosat, Natural Hazards and Earth System Sciences 18 (2018) 515–529.

K. Rasul, C. Seward, I. Schuster, R. Vollgraf, Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting, in: International conference on machine learning, PMLR, pp. 8857–8868.

K. Rasul, A.-S. Sheikh, I. Schuster, U. M. Bergmann, R. Vollgraf, Multivariate probabilistic time series forecasting via conditioned normalizing flows, in: International Conference on Learning Representations.

S. I. Resnick, Tail equivalence and its applications, Journal of Applied Probability 8 (1971) 136–156.

S. I. Resnick, Extreme Values, Regular Variation, and Point Processes, Springer, New York, 1987.

J. Richards, R. Huser, Regression modelling of spatiotemporal extreme us wildfires via partially-interpretable neural networks, arXiv preprint arXiv:2208.07581 (2022).

J. Richards, R. Huser, E. Bevacqua, J. Zscheischler, Insights into the drivers and spatiotemporal trends of extreme mediterranean wildfires with statistical deep learning, Artificial Intelligence for the Earth Systems 2 (2023) e220095.

J. Richards, M. Sainsbury-Dale, A. Zammit-Mangion, R. Huser, Neural bayes estimators for censored inference with peaks-over-threshold models, Journal of Machine Learning Research 25 (2024) 1–49.

A. Riebler, S. H. Sørbye, D. Simpson, H. Rue, An intuitive bayesian spatial model for disease mapping that accounts for scaling, Statistical methods in medical research 25 (2016) 1145–1165.

Ó. R. de Rivera, J. Espinosa, J. Madrigal, M. Blangiardo, A. López-Quílez, Spatio-temporal marked point process model to understand forest fires in the mediterranean basin, Journal of Agricultural, Biological and Environmental Statistics (2024) 1–30.

H. Robbins, S. Monro, A stochastic approximation method, The annals of mathematical statistics (1951) 400–407.

G. O. Roberts, J. S. Rosenthal, Optimal scaling for various Metropolis-Hastings algorithms, Statistical Science 16 (2001).

H. Rootzén, J. Segers, J. L. Wadsworth, Multivariate peaks over thresholds models, Extremes 21 (2018a) 115–145.

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

H. Rootzén, N. Tajvidi, Multivariate generalized pareto distributions, Bernoulli 12 (2006) 917–930.

H. Rue, S. Martino, N. Chopin, Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations, Journal of the Royal Statistical Society Series B: Statistical Methodology 71 (2009) 319–392.

D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, nature 323 (1986) 533–536.

D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, DeepAR: Probabilistic forecasting with autoregressive recurrent networks, International journal of forecasting 36 (2020) 1181–1191.

D. Santoro, T. Ciano, M. Ferrara, A comparison between machine and deep learning models on high stationarity data, Scientific Reports 14 (2024) 19409.

R. Schmidt, U. Stadtmüller, Non-parametric estimation of tail dependence, Scandinavian journal of statistics 33 (2006) 307–335.

J. Segers, Max-stable models for multivariate extremes, arXiv preprint arXiv:1204.0332 (2012).

L. S. Shapley, A value for n-person games, in: H. W. Kuhn, A. W. Tucker (Eds.), Contributions to the Theory of Games II, Princeton University Press, Princeton, 1953, pp. 307–317.

M. Sibuya, Bivariate extreme statistics, I, Annals of the Institute of Statistical Mathematics 11 (1960) 195–210.

M. Sibuya, et al., Bivariate extreme statistics, Annals of the Institute of Statistical Mathematics 11 (1960) 195–210.

D. Simpson, H. Rue, A. Riebler, T. G. Martins, S. H. Sørbye, Penalising model component complexity: A principled, practical approach to constructing priors, Statistical Science 32 (2017).

J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: International conference on machine learning, pmlr, pp. 2256–2265.

M. L. Stein, A parametric model for distributions with flexible behavior in both tails, Environmetrics 32 (2021) e2658.

D. Sun, J. O. Berger, Objective bayesian analysis for the multivariate normal model, Bayesian Statistics 8 (2007) 525–562.

H. Tabari, Extreme value analysis dilemma for climate change impact assessment on global flood and extreme precipitation, Journal of Hydrology 593 (2021) 125932.

A. Tancredi, C. Anderson, A. O'Hagan, Accounting for threshold uncertainty in extreme value estimation, Extremes 9 (2006) 87–106.

A. N. Thiombiano, S. El Adlouni, A. St-Hilaire, T. B. Ouarda, N. El-Jabi, Nonstationary frequency analysis of extreme daily precipitation amounts in Southeastern Canada using a peaks-over-threshold approach, Theoretical and Applied Climatology 129 (2017) 413–426.

M. M. Tibbits, C. Groendyke, M. Haran, J. C. Liechty, Automated factor slice sampling, Journal of Computational and Graphical Statistics 23 (2014) 543–563.

M. Tonini, M. G. Pereira, J. Parente, C. Vega Orozco, Evolution of forest fires in portugal: from spatio-temporal point events to smoothed density maps, Natural Hazards 85 (2017) 1489–1510.

J. Van Niekerk, E. Krainski, D. Rustand, H. Rue, A new avenue for bayesian inference with inla, Computational Statistics & Data Analysis 181 (2023) 107692.

A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, P.-C. Bürkner, Rank-Normalization, Folding, and Localization: An Improved $\widehat{R}$ for Assessing Convergence of MCMC (with Discussion), Bayesian Analysis 16 (2021) 667 – 718.

J. L. Wadsworth, R. Campbell, Statistical inference for multivariate extremes via a geometric approach, Journal of the Royal Statistical Society Series B: Statistical Methodology 86 (2024) 1243–1265.

S. Watanabe, M. Opper, Asymptotic equivalence of bayes cross validation and widely applicable information criterion in singular learning theory., Journal of machine learning research 11 (2010).

S. Westra, H. J. Fowler, J. P. Evans, L. V. Alexander, P. Berg, F. Johnson, E. J. Kendon, G. Lenderink, N. Roberts, Future changes to the intensity and frequency of short-duration extreme rainfall, Reviews of geophysics 52 (2014) 522–555.

S. Wi, J. B. Valdés, S. Steinschneider, T.-W. Kim, Non-stationary frequency analysis of extreme precipitation in South Korea using peaks-over-threshold and annual maxima, Stochastic environmental research and risk assessment 30 (2016) 583–606.

C. K. Wikle, A. Zammit-Mangion, Statistical deep learning for spatial and spatiotemporal data, Annual Review of Statistics and Its Application 10 (2023) 247–270.

H. C. Winter, J. A. Tawn, S. J. Brown, Modelling the effect of the el niÑo-southern oscillation on extreme spatial temperature events over australia, The Annals of Applied Statistics 10 (2016) 2075–2101.

D. H. Wolpert, Stacked generalization, Neural networks 5 (1992) 241–259.

D. G. Woolford, D. L. Martell, C. B. McFayden, J. Evens, A. Stacey, B. M. Wotton, D. Boychuk, The development and implementation of a human-caused wildland fire occurrence prediction system for the province of ontario, canada, Canadian Journal of Forest Research 51 (2021) 303–325.

D. D. Xi, S. W. Taylor, D. G. Woolford, C. Dean, Statistical models of key components of wildfire risk, Annual Review of Statistics and Its Application 6 (2019) 197–222.

H. Xu, F. P. Schoenberg, Point process modeling of wildfire hazard in los angeles county, california, The Annals of Applied Statistics 5 (2011) 684–704.

R. Yadav, R. Huser, T. Opitz, Spatial hierarchical modeling of threshold exceedances using rate mixtures, Environmetrics 32 (2021) e2662.

R. Yadav, R. Huser, T. Opitz, L. Lombardo, Joint modelling of landslide counts and sizes using spatial marked point processes with sub-asymptotic mark distributions, Journal of the Royal Statistical Society Series C: Applied Statistics 72 (2023) 1139–1161.

P. de Zea Bermudez, M. Amaral Turkman, K. Turkman, A predictive approach to tail probability estimation, Extremes 4 (2001) 295–314.

H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI conference on artificial intelligence, volume 35, pp. 11106–11115.