Fast amortised inference for spatial extremes using neural Bayes estimators

Jordan Richards

School of Mathematics and Maxwell Institute for Mathematical Sciences,
University of Edinburgh, UK

STSDS 2025

Collaborators - Methodology

Plus Raphaël Huser (KAUST) and Andrew Zammit-Mangion (UoW)

Collaborators - Comparative case study

Preface

Fast amortised inference for spatial extremes using neural Bayes estimators

Preface

This talk is not for...

Fast amortised inference for spatial extremes using neural Bayes estimators

Preface

This talk is not for...

Fast amortised inference for spatial extremes using neural Bayes estimators

Preface

This talk is not for...

Fast amortised inference for spatial extremes using neural Bayes estimators

Preface

This talk is not for...

Fast amortised inference for spatial extremes using neural Bayes estimators

Preface

This talk **is** for...

Fast amortised inference for spatial extremes using neural Bayes estimators

Specifically, anyone who estimates models that take a bit **too long** to fit or require **repeated fits**, e.g., online, bootstrap.

Likelihood-based inference

- Statistical inference typically proceeds via the **likelihood function**.
- However, the likelihood function may be
  - **unavailable** (e.g., implicit generative/simulator models), or
  - **computationally intractable** (e.g., max-stable processes, censored likelihoods).
- One may **approximate** the likelihood function (e.g., composite likelihood, the Vecchia approximation, etc.), but this involves a trade-off between computational and statistical efficiency.
- Alternatively, one may use **likelihood-free inference**, e.g., ABC or neural estimators.

Neural estimators

- A **neural estimator** $\hat{\theta}(\mathbf{Z})$ is a neural network that takes in data $\mathbf{Z}$ as input and provides a parameter point estimate $\boldsymbol{\theta}$ as an output. See, e.g., Lenzi et al. (2023).

- Their construction is simple:
  - Sample (many) parameter vectors $\boldsymbol{\theta}$ from a prior $\Pi(\cdot)$.
  - Simulate $\boldsymbol{Z}$ from the model, conditional on these parameters.
  - Train a neural network that maps the simulated data $\boldsymbol{Z} \mapsto \boldsymbol{\theta}$ to the true parameters by minimising some loss function $L(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}(\mathbf{Z}))$.

Lenzi, A., Bessac, J., Rudi, J., & Stein, M. L. (2023). Neural networks for parameter estimation in intractable models. Computational Statistics & Data Analysis, 185, 107762.

Bayes estimators

Connecting neural estimators to classical estimators?

- A non-negative loss function, $L(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}(\boldsymbol{Z}))$, assesses an estimator, $\widehat{\boldsymbol{\theta}}(\cdot)$, for a single parameter vector, $\boldsymbol{\theta}$, and model realisation, $\boldsymbol{Z}$.

- The Bayes risk averages the loss function over sample space $\mathcal{Z} \subseteq \mathbb{R}^n$ and parameter space $\Theta \subseteq \mathbb{R}^p$ with respect to the prior, $\Pi(\cdot)$;

$$r(\widehat{\boldsymbol{\theta}}(\cdot)) = \int_{\Theta} \left[\int_{\mathcal{Z}} L(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}(\boldsymbol{z})) f(\boldsymbol{z} \mid \boldsymbol{\theta}) \mathrm{d}\boldsymbol{z} \right] \mathrm{d}\Pi(\boldsymbol{\theta}),$$

  where $f(\mathbf{z} \mid \boldsymbol{\theta})$ is the probability density function of the data conditional on $\boldsymbol{\theta}$.

- A minimiser of the Bayes risk is said to be a Bayes estimator with respect to $L(\cdot, \cdot)$ and $\Pi(\cdot)$.

Neural Bayes estimators

- Denote a neural estimator by $\widehat{\boldsymbol{\theta}}_{\boldsymbol{\gamma}}(\cdot)$, where $\boldsymbol{\gamma}$ is a vector of neural-network parameters ("weights" and "biases").

- A neural estimator is trained by solving the optimisation task,

$$\boldsymbol{\gamma}^* = \arg\min_{\boldsymbol{\gamma}} \frac{1}{K} \sum_{k=1}^{K} L(\boldsymbol{\theta}^{(k)}, \widehat{\boldsymbol{\theta}}_{\boldsymbol{\gamma}}(\boldsymbol{Z}^{(k)})), \tag{1}$$

  where $\boldsymbol{\theta}^{(k)}$, $k = 1, \dots, K$, is sampled from the prior $\Pi(\cdot)$ and, for each k, data $\boldsymbol{Z}^{(k)}$ are sampled from $f(\cdot \mid \boldsymbol{\theta}^{(k)})$.

- Since the objective function in (1) is a Monte Carlo approximation of the Bayes risk, neural estimators approximate the Bayes estimator.

Neural Bayes estimators

- A neural Bayes estimator $\widehat{\boldsymbol{\theta}}_{\boldsymbol{\gamma}^*}(\cdot)$ approximately inherits the attractive properties of Bayes estimators (e.g., consistency, asymptotic efficiency). See Sainsbury-Dale et al. (2024).
- The loss function specifies the Bayes estimator and, hence, the neural Bayes estimator (NBE).
  - Under the absolute-error loss, the NBE approximates the posterior median.
  - Under the squared-error loss, the NBE approximates the posterior expectation.
  - Under the tilted loss, $(\hat{\theta} - \theta)(\mathbb{I}(\hat{\theta} - \tau))$, the NBE approximates the posterior τ-quantile.
  - Weighted tilted losses can be used to get conservative return level estimates. See, e.g., Richards et al. (2025).

Sainsbury-Dale, M., Zammit-Mangion, A., & Huser, R. (2024). Likelihood-free parameter estimation with neural Bayes estimators. *The American Statistician*, 78(1), 1-14.

Richards, J., Alotaibi, N., Cisneros, D., Gong, Y., Guerrero, M. B., Redondo, P., & Shao, X. (2025). Modern extreme value statistics for Utopian extremes. *Extremes*, 28 (1), 149-171.

Uncertainty Quantification

Performing principled, fast uncertainty quantification?

- It can be shown that the Bayes estimator under the loss

$$L(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \sum_{k=1}^{p} (\hat{\theta}_k - \theta_k)(\mathbb{I}(\hat{\theta}_k - \tau)), \quad \tau \in (0, 1), \tag{2}$$

  is the vector of marginal posterior τ-quantiles.

- We can chain together loss functions and build a NBE that targets multiple quantiles, e.g., credible interval estimation.

- When approximating multiple quantiles (e.g., to construct credible intervals), the neural-network architecture can be designed to prevent quantile crossing.

Sainsbury-Dale, M., Zammit-Mangion, A., **Richards, J.**, & Huser, R. (2025). Neural Bayes estimators for irregular spatial data using graph neural networks. *JCGS*, 34(3), 1153-1168

Neural Bayes estimators for replicated data

Accounting for estimation with replicated data?

Proposition

Assume that, for some loss function $L(\cdot, \cdot)$ and prior distribution $\Pi(\cdot)$, the Bayes estimator exists and is unique. If the data $\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_m$ are conditionally independent given $\boldsymbol{\theta}$, then the Bayes estimator is permutation invariant. That is,

$$\widehat{\boldsymbol{\theta}}_{\mathrm{Bayes}}(\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_m) = \widehat{\boldsymbol{\theta}}_{\mathrm{Bayes}}(\boldsymbol{Z}_{\pi(1)}, \dots, \boldsymbol{Z}_{\pi(m)})$$

for any permutation $\pi(\cdot)$.

Neural Bayes estimators for replicated data

- To ensure permutation invariance, we construct our neural estimator with permutation-invariant neural networks.

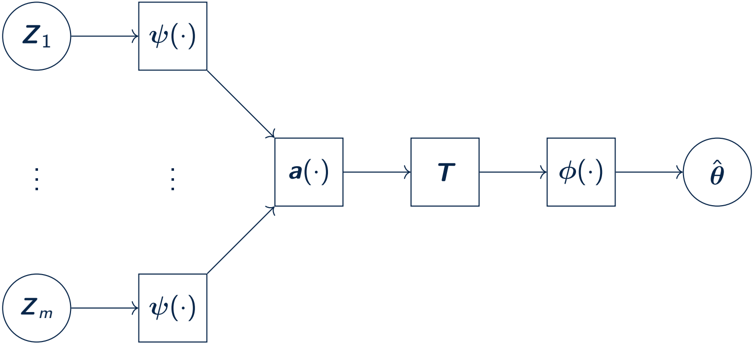
- Specifically, we use the DeepSets framework (Zaheer et al., 2017),

$$\widehat{\boldsymbol{\theta}}(\boldsymbol{Z}) = \boldsymbol{\phi}(\boldsymbol{a}(\{\boldsymbol{\psi}(\boldsymbol{Z}_i)\}_{i=1,\ldots,m})),$$

  with $\boldsymbol{\psi} : \mathbb{R}^n \to \mathbb{R}^w$ and $\boldsymbol{\phi} : \mathbb{R}^w \to \mathbb{R}^p$ generic neural networks, and $\boldsymbol{a}(\cdot)$ a permutation-invariant aggregation function.

Neural Bayes estimators for replicated data

Schematic of a neural Bayes estimator based on the DeepSets framework:



The neural network $\phi(\cdot)$ is densely-connected (vanilla).

Types of neural networks

Choose $\psi(\cdot)$ based on the modality of $\mathbf{Z}$:

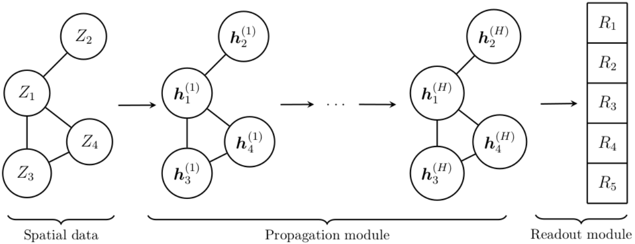
- Dense neural networks (DNNs) can be used for univariate or multivariate data, but do not exploit structure in $\mathbf{Z}$.
- Convolutional neural networks (CNNs):
  - Extract spatial patterns in data.
  - Require data to be measured on a fully observed, regular grid.
  - Can only be used with grids of a single size.

Irregular spatial data

- We propose to construct neural Bayes estimators for irregular spatial data using graph neural networks (GNNs; e.g., Wu et al., 2021).
  - The spatial data are viewed as a graph with edges weighted by spatial distance.

- GNNs:
  - can be applied to data measured over irregular spatial locations,
  - explicitly model spatial dependence by generalising the convolution operation in CNNs to graphical data (making them parsimonious), and
  - are not tied to a specific set of spatial locations, so the expensive training stage need only be performed once for a given spatial model.

Irregular spatial data



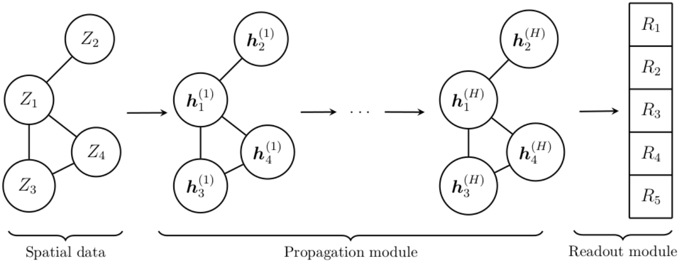
For $\boldsymbol{h}_j^{(l)}$, $l = 1, \ldots, H$, a hidden-feature vector at location $\boldsymbol{s}_j$, $\boldsymbol{h}_j^{(0)} = Z_j$:

$$\text{Propagation Module}: \quad \boldsymbol{h}_j^{(l)} = g\left(\boldsymbol{\Gamma}_1^{(l)} \boldsymbol{h}_j^{(l-1)} + \boldsymbol{\Gamma}_2^{(l)} \bar{\boldsymbol{h}}_j^{(l)} + \boldsymbol{b}^{(l)}\right)$$

$$\bar{\boldsymbol{h}}_j^{(l)} = \sum_{j' \in \mathcal{N}(j)} \boldsymbol{w}_j^{(l)}(\boldsymbol{s}_j, \boldsymbol{s}_{j'}; \boldsymbol{\zeta}^{(l)}) \odot \boldsymbol{\rho}^{(l)}(\boldsymbol{h}_j^{(l-1)}, \boldsymbol{h}_{j'}^{(l-1)}; \boldsymbol{\varphi}^{(l)}),$$

for hyperparameters, activation $g(\cdot)$, neighbourhood $\mathcal{N}(\cdot)$, and learnable $\boldsymbol{\rho}^{(l)}(\cdot, \cdot)$.

Irregular spatial data



For $\boldsymbol{h}_j^{(l)}$, $l = 1, \ldots, H$, a hidden-feature vector at location $\boldsymbol{s}_j$, $\boldsymbol{h}_j^{(0)} = Z_j$, $\boldsymbol{u}(\cdot)$ a set aggregation function (e.g., elementwise mean):

$$\text{Readout Module}: \quad \boldsymbol{R} \quad = \boldsymbol{u}(\{\boldsymbol{h}_j^{(H)} : j = 1, \ldots, n\}).$$

Training for varying spatial configurations

- We treat the spatial locations S as a random point pattern belonging to the space $\mathcal{S}$ of all possible spatial configurations.
- The Bayes risk is then

$$r(\widehat{\boldsymbol{\theta}}(\cdot,\cdot)) = \int_{\Theta}\int_{\mathcal{S}}\int_{\mathcal{Z}_S} L(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}(\boldsymbol{Z}, S)) f(\boldsymbol{Z} \mid \boldsymbol{\theta}, S) \mathrm{d}\boldsymbol{Z} \mathrm{d}\Omega(S) \mathrm{d}\Pi(\boldsymbol{\theta}), \quad (3)$$

where $\mathcal{Z}_S \subseteq \mathbb{R}^{|S|}$ and $\Omega(\cdot)$ is a distribution for S.

- We then solve the empirical risk minimisation problem:

$$\gamma^* \approx \underset{\gamma}{\arg\min} \frac{1}{K} \sum_{k=1}^{K} L(\boldsymbol{\theta}^{(k)}, \widehat{\boldsymbol{\theta}}_\gamma(\boldsymbol{Z}^{(k)}, S^{(k)})), \quad (4)$$

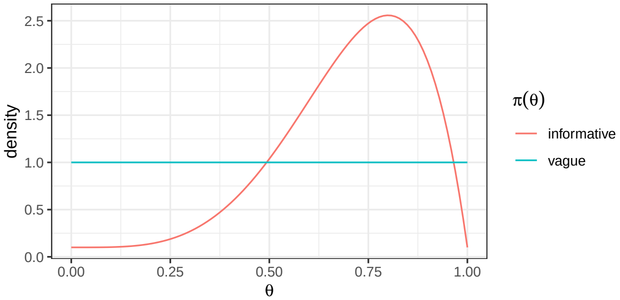
where $S^{(k)} \sim \Omega(S)$.

Training for varying spatial configurations

Theorem

Assume that the Bayes estimate $\hat{\boldsymbol{\theta}}^{\star}$ has finite posterior expected loss $\int_{\Theta} L(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{\star}) p(\boldsymbol{\theta} \mid \boldsymbol{Z}, S) \mathrm{d}\boldsymbol{\theta}$ for all fixed $\boldsymbol{Z} \in \mathcal{Z}_S \subseteq \mathbb{R}^{|S|}$ and $S \in \mathcal{S}$. If S and $\boldsymbol{\theta}$ are independent, then the Bayes estimator $\hat{\boldsymbol{\theta}}^{\star}(\boldsymbol{Z}, S)$ is invariant to the distribution $\Omega(\cdot)$ of S among all strictly positive measures.

Training for varying spatial configurations

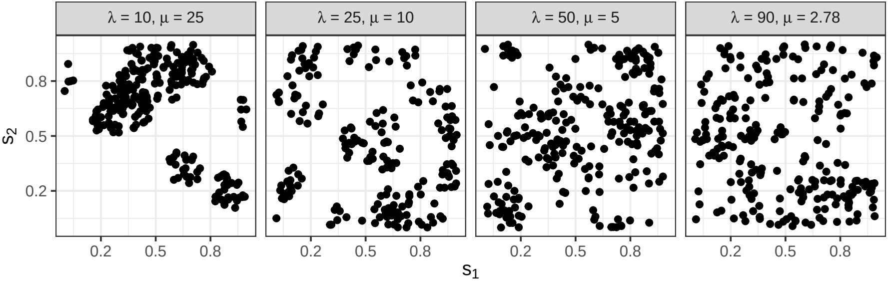


- The parameter prior $\Pi(\boldsymbol{\theta})$ directly influences the Bayes estimator.

- This is not the case for the distribution $\Omega(S)$: it doesn't matter if $\Omega(S)$ is informative or vague, the Bayes estimator is the same!

Training for varying spatial configurations

- We simulate locations $S^{(k)}$ from a Matérn cluster process (with varying intensity) during training.



Simulation study: Gaussian process

- Matérn Gaussian process with 2 parameters to estimate: measurement error standard deviation $\sigma_\epsilon > 0$ and range $\rho > 0$ (fixed smoothness $\nu = 1$).

- We use the priors $\sigma_\epsilon \sim \mathrm{Unif}(0, 1)$ and $\rho \sim \mathrm{Unif}(0.05, 0.5)$. The total training time is 24 minutes.

- We compare our estimator to the maximum-a-posteriori (MAP) estimator.

Simulation study: Gaussian process

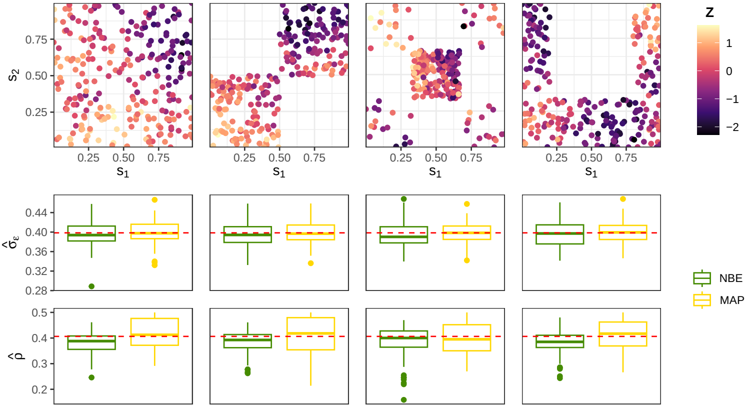


Figure: Several spatial data sets (top row) and empirical marginal sampling distributions (second and third row) of two estimators for a Gaussian Process model under a single parameter configuration (red dashed line).

Simulation study: Gaussian process

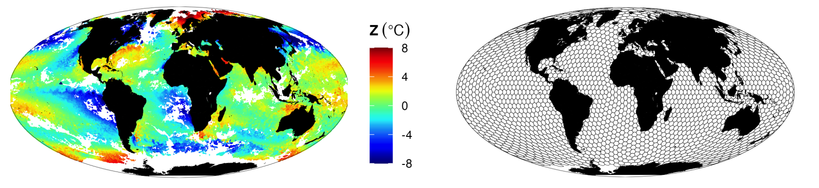
- The neural Bayes estimator has similar RMSE to the MAP estimator (0.054 and 0.046, respectively).

- The MAP estimator takes 1.2 seconds to estimate the parameters from a single data set, while the neural Bayes estimator is substantially faster, taking only 0.002 seconds (a 600-fold speedup).

- The empirical coverages for σ_ϵ and ρ using our quantile networks targeting the 2.5 and 97.5 percentiles are 94.6% and 95.2%, respectively, which are close to the nominal value.

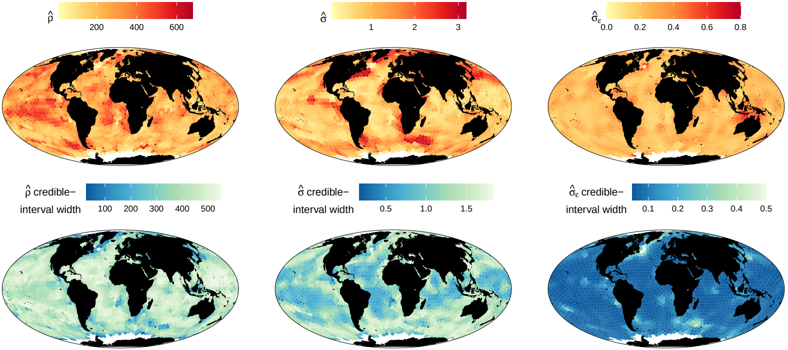
Application example

- Sea-surface temperature data obtained from the Suomi National Polar-orbiting Partnership (NPP) weather satellite.
- To deal with nonstationarity, we use a local moving-window approach where we fit a Matérn Gaussian process model using detrended data from within a given region and its neighbouring regions (Haas, 1990).



Application example

- Fitting 2161 Matérn Gaussian process models using our neural Bayes estimator required three minutes on a single GPU – this included generation of 95% credible intervals!

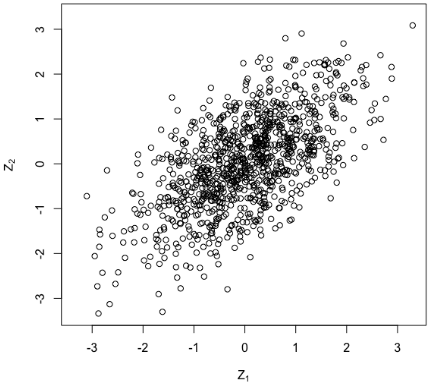


Background: peaks-over-threshold models

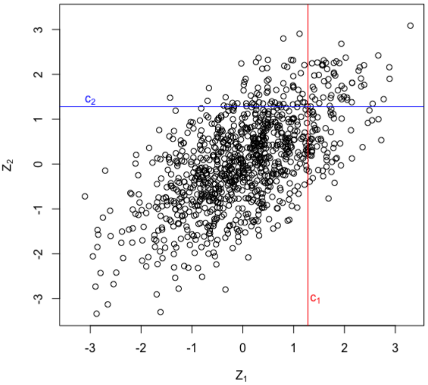
- When doing inference for extremal dependence, we might actually **choose to treat our data as censored!**
- Likelihood estimators for spatial extremal dependence models are typically highly biased if spatial extreme events include marginally non-extreme values (Huser et al., 2016);
- Can be mitigated in a peaks-over-threshold framework:
  - Impose **artificial censoring** of our data during inference;
  - Remove contribution of **non-extreme** values to the likelihood;
  - Extremity determined by some high censoring threshold, e.g., the τ-quantile for τ close to one.

Huser, R., Davison, A. C., and Genton, M. G. (2016). Likelihood estimators for multivariate extremes. *Extremes*, 19:79–103.

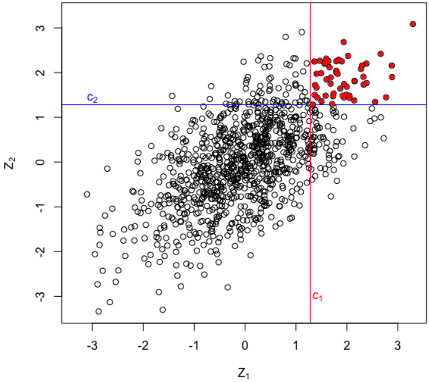
Background: peaks-over-threshold models



Background: peaks-over-threshold models

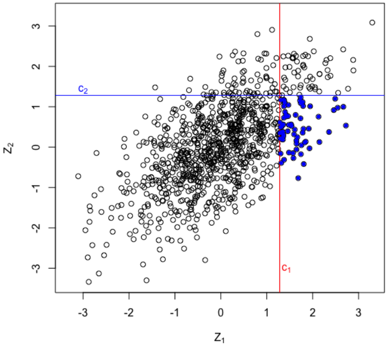


Background: peaks-over-threshold models



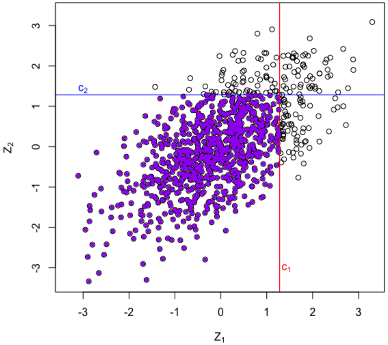
Both components extreme $\Rightarrow$ **both fully observed**.
Likelihood contribution of (Z_1, Z_2): $f(z_1, z_2)$

Background: peaks-over-threshold models



Only Z_1 is extreme $\Rightarrow$ Z_2 **treated as left censored at** c_2.
Record this in $\mathcal{I} = \mathbb{1}\{Z_2 < c_2\}$.
Likelihood contribution of $(Z_1, \mathcal{I})$: $\int_{-\infty}^{c_2} f(z_1, z_2)\mathrm{d}z_2$.

Background: peaks-over-threshold models



Record $\mathcal{I}_1 = \mathbb{1}\{Z_1 < c_1\}$ and $\mathcal{I}_2 = \mathbb{1}\{Z_2 < c_2\}$.
Likelihood contribution of $(\mathcal{I}_1, \mathcal{I}_2)$: $\int_{-\infty}^{c_1}\int_{-\infty}^{c_2} f(z_1, z_2)\mathrm{d}z_1\mathrm{d}z_2$.
The exact values of (Z_1, Z_2) are irrelevant!

Background: peaks-over-threshold models

- **Extends naturally to D-dimensions**;
- The contribution of an observation to the **censored-likelihood** is a C-variate integral, where $C \leq D$ is the number of censored values;
- Likely to be **intractable** for any $C > 0$ and **expensive** for large C;
- We adapt neural Bayes estimators so that they **mimic** peaks-over-threshold inference;
- Note: the censoring scheme is **chosen a priori**. This is not random censoring or missing-at-random. For incomplete data, see Sainsbury-Dale et al. (2025a)

Sainsbury-Dale, M., Zammit-Mangion, A., Cressie, N., and Huser, R. (2025). Neural Parameter Estimation with Incomplete Data. *arXiv:2501.04330*

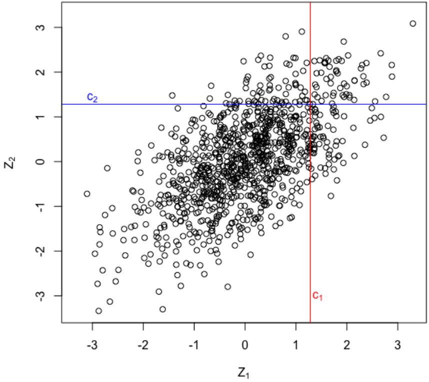
Defining censored inputs

Adapting NBEs for estimation with censored data?

- To the neural estimator, we supply data $(\mathbf{Z}, \mathcal{I})$, where $\mathcal{I}$ is a one-hot encoded vector of **components with censoring**.
- For likelihood-based inference, we reduce the contribution of censored values, to estimation of $\boldsymbol{\theta}$, by integrating them out of $f(\cdot)$.
- For our neural estimator, we instead set censored values to a fixed constant outside of the support of $\mathbf{Z}$.

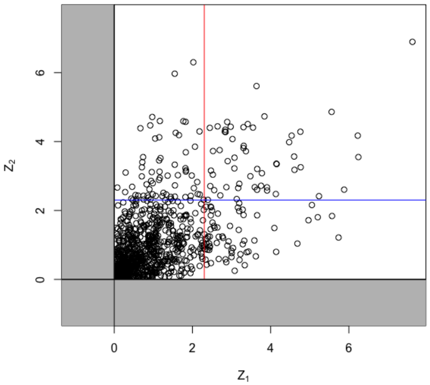
NBE input specification

We first transform $\mathbf{Z} \mapsto \mathbf{Z}^*$ onto **standard margins with a finite lower-endpoint** (does not alter the dependence structure in $\mathbf{Z}$).



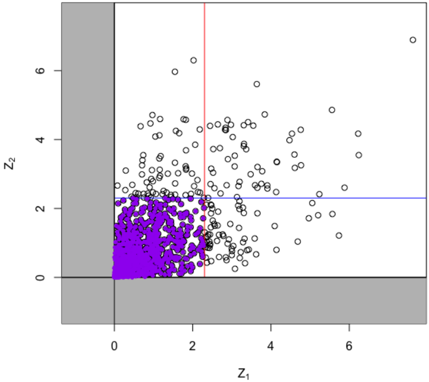
NBE input specification

To solve ii), we first transform $\mathbf{Z} \mapsto \mathbf{Z}^*$ onto **standard margins with a finite lower-endpoint** (does not alter the dependence structure in $\mathbf{Z}$).



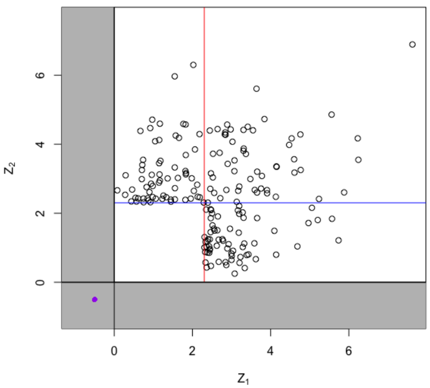
NBE input specification (cont.)

We **then set "censored values"** to a constant c^* outside of the support for $\mathbf{Z}^*$...



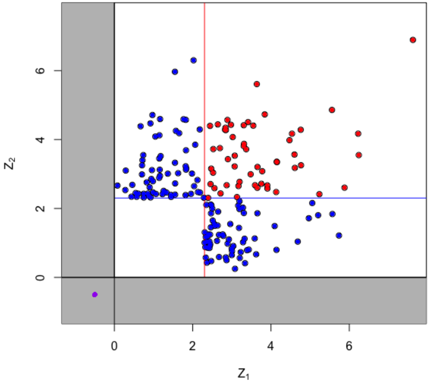
NBE input specification (cont.)

...removing information about their **exact values**.



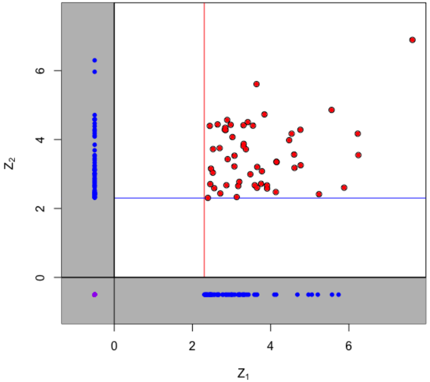
NBE input specification (cont.)

If c^* is outside of the support for $\mathbf{Z}^*$, then the NBE will not **mistake it for an uncensored value**.



NBE input specification (cont.)

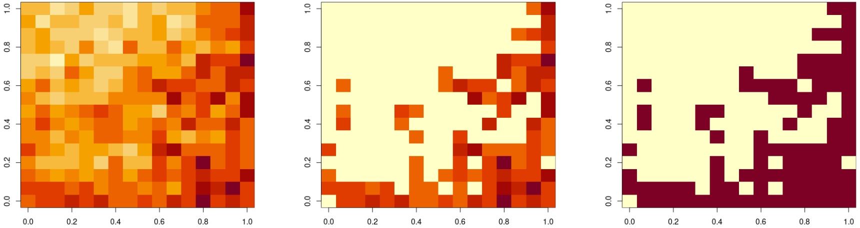
Information about **extreme** components is retained and will continue to contribute to estimation of $\boldsymbol{\theta}$.



New input

Our NBE is then trained on $(\mathbf{Z}^*, \mathcal{I})$ and a user can perform a similar transformation of their own data before supplying it to the NBE to get parameter estimates.

Left: Realisation $\mathbf{Z}$ from a max-stable process.
Centre: $\mathbf{Z}^*$ with $\tau = 0.9$ censoring and $c^* = 0$.
Right: one-hot encoding $\mathcal{I}$.



Models

We consider inference with 3 **popular models**:

- Max-stable process (MSP) and inverted MSP (1/MSP),
- Random scale mixture (Huser and Wadsworth, 2019),

$$\{Z(\mathbf{s})\} = R^{\delta}\{W(\mathbf{s})^{1-\delta}\},$$

  where W is a standard Matérn Gaussian process with the same margins as the heavy-tailed r.v. R and $\delta \in [0, 1]$;
- If $\delta \geq 1/2$, then $Z(\cdot)$ is **asymptotically dependent.**

Asymptotic dependence: $\chi = \lim_{q \to 1} \Pr[F_1\{Z(\mathbf{s}_1)\} > q \mid F_2\{Z(\mathbf{s}_2)\} > q]$.

Huser, R. and Wadsworth, J. L. (2019). Modeling spatial processes with unknown extremal dependence class. *JASA. 114(525):434–444*

Simulation study 1: outline

- Consider MSP and IMSP (1/MSP) with $\tau = 0.9$;
- Both have **range** $\lambda > 0$ and **smoothness** $\kappa \in (0, 2]$, with unif. priors;
- Simulate 200 replicates on a 16×16 grid;
- Compare to the **competing likelihood-based approach**, i.e., censored pairwise-likelihood (cPL);
- cPL(∞): all pairs; cPL(3), only those within 3 units.

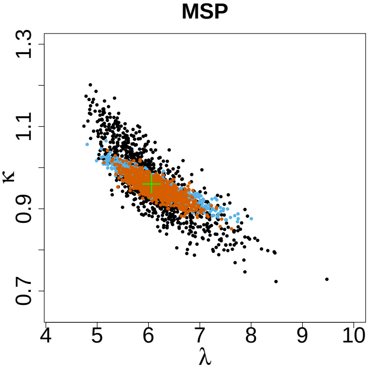
Simulation study 1: results

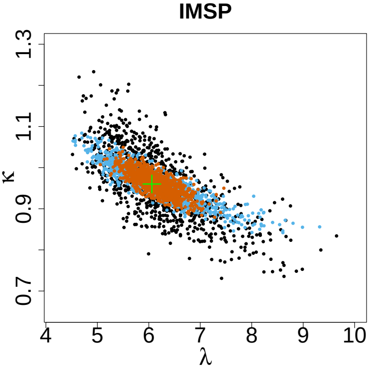
Marginal test risk (s.d.) evaluated on 1000 **test parameter sets**.

	MSP		IMSP	
	λ	κ	λ	κ
NBE	**2.4 (0.1)**	**1.8 (0.1)**	**2.6 (0.1)**	**2.2 (0.1)**
cPL (3)	3.5 (0.1)	2.2 (0.1)	4.6 (0.2)	3.2 (0.1)
cPL (∞)	4.3 (0.1)	6.4 (0.2)	5.4 (0.2)	6.8 (0.2)

Simulation study 1: joint distribution

- Empirical joint dist. of estimators with single true vector $\boldsymbol{\theta}$;
- Black: cPL(∞). Blue: cPL(3). Brown: NBE.
- **NBE captures well the joint distribution, but with lower variance than the competing likelihood approach.**



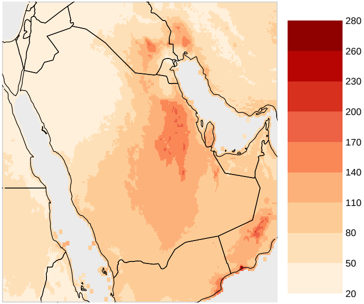


Simulation study 1: conclusion

- Takeaways:
  - NBE gives **large improvements in statistical efficiency**;
  - **Improvements in computational efficiency**!
    NBE takes exactly 0.0016 seconds; cPL takes $\approx$ 2 to 10 minutes.
- We showcase similar for r-**Pareto, Gaussian, and HW processes.**
- These NBEs are now ready-to-ship! Anyone with **new data** observed on a similar grid can immediately get parameter estimates in milliseconds...but only if they use $\tau = 0.9$.
- We can train an estimator for a general τ if we **supply τ to the estimator** as an input.

Application

Application to monthly Saudi Arabian $PM_{2.5}$ (Van Donkelaar et al., 2021) concentrations shows the computational gains of our amortised estimator.



Observation of surface average $PM_{2.5}$ conc. (μg/m^3) for Jul. 2012.

Van Donkelaar, A., et al. (2021). Monthly global estimates of fine particulate matter and their uncertainty. *Environmental Science & Technology*, 55(22):15287–15300.

Application (cont.)

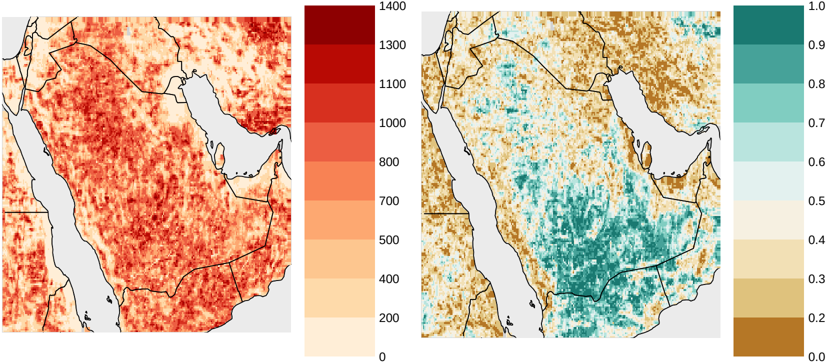
- Data are arranged on a 242×182 regular grid; monthly, 1998–2020.
- Fit local anisotropic HW processes with $\tau = 0.9$ (five params.);
- To all possible subsets of data on $G \times G$ grids for smoothing level $G \in \{4, 8, 16, 24, 32\}$. This is over 130,000 fits!
- Once an estimator is trained (roughly 24 to 72 hours), a single model fit takes between 1 and 4 milliseconds to estimate.

Application (cont.)

- Data are arranged on a 242×182 regular grid; monthly, 1998–2020.
- Fit local anisotropic HW processes with $\tau = 0.9$ (five params.);
- To all possible subsets of data on $G \times G$ grids for smoothing level $G \in \{4, 8, 16, 24, 32\}$. This is over 130,000 fits!
- Once an estimator is trained (roughly 24 to 72 hours), a single model fit takes between 1 and 4 milliseconds to estimate.
- Speed-up/dimension comparison:
  - Full censored likelihood-based inference is limited to $D \approx 6^2 = 36$ and takes roughly 12 hours per estimate;
  - NBE with $D = 32^2 = 1024$ and $\approx$ **10 million times faster.**

Results

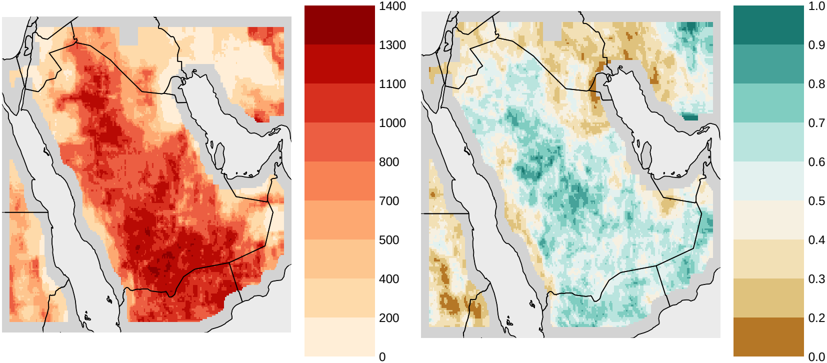
Each pixel is a single model fit.



λ (left) and δ (right) estimates for $G = 4$.

Results (cont.)

Each pixel is a single model fit.



λ (left) and δ (right) estimates for $G = 16$.

Application (cont.)

- We can also perform parameter uncertainty assessment for free, with 1000 bootstrap estimates obtained within seconds;
- In total, our analysis uses 130 million model fits...
- **...which is far more than any comparable application[1]!**
- And only five estimators have been trained (one for each G).

[1] as far as we know.

Inspired By Heaton et al. (2019)

- **Focus:** Block maxima (e.g., annual/seasonal maxima at each location).
- **Methods:** Scalable spatial extremes models
  - Low-rank & sparse covariance/precision matrices
  - Neural Bayes Estimation
  - Semi-parametric quantile regression (SPQR)
- **Goal:** Compare speed, accuracy, and inference of extremal dependence.

Heaton, M. J., et al. (2019). A case study competition among methods for analyzing large spatial data. *JABES*, 24:398–425.

Random scale Mixture

Random Scale Mixture (Huser and Wadsworth, 2019): Unified framework permitting smooth transitions between dependence classes.

$$X(\boldsymbol{s}, t) = R(t)^{\delta} W(\boldsymbol{s})^{1-\delta}$$

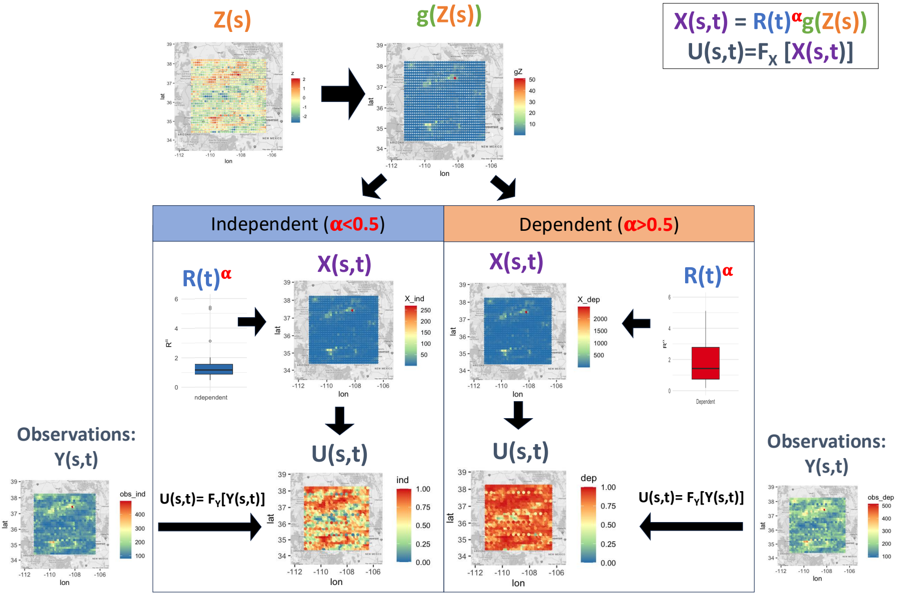
- Challenge: MCMC mixing due to $R(t), W(\boldsymbol{s}) \in (1, \infty)$ (Zhang et al., 2022)

Our Random Scale Mixture Model:

$$X(\boldsymbol{s}, t) = R(t)^{\alpha} g(Z(\boldsymbol{s})),$$

- Scaling Factor: $R(t) \sim \text{Lévy}(m, 1/2)$ where $m \in \mathbb{R}$.
- Gaussian Process: $Z(\boldsymbol{s}) \sim GP(\mathbf{0}, \mathbf{K}_{\Theta})$ with correlation function $\mathbf{K}_{\Theta}$.
- Transformation: $g(\cdot) = \{1 - \Phi(\cdot)\}^{-1} - 1$
  - Standard normal $\rightarrow$ shifted Pareto.
- Dependence parameter: $\alpha \in (0, 1)$
  - Asymp. independence if $\alpha \leq 0.5$ and dependence if $\alpha > 0.5$.

Scale Mixture Models: An Illustration



Scalable Methods: An Overview

Gold Standard: Model full Gaussian process without approximations

Approximate Likelihood Methods:

1. **Low-rank Methods:** Replace $\mathbf{C}_\Theta$ with low-rank $\mathbf{\Phi K \Phi}' + \tau^2 \mathbf{I}$ /(Cressie et al., 2022).
2. **Covariance Tapering :** Compactly supported correlation function $\tilde{\mathbf{C}}_\Theta$ (Furrer et al., 2006).
3. **Vecchia Approximations:** Sparse Cholesky factorization of precision matrix $\mathbf{Q}_\Theta = \mathbf{C}_\Theta^{-1} \approx \mathbf{LL}'$ based on conditioning sets/neighborhoods (Vecchia, 1988; Katzfuss et al., 2020).
4. **Semi-parametric quantile regression (SPQR):** Train a neural network to map parameters and data to a surrogate likelihood function (Majumder et al., 2024).

Likelihood-free Inference:

1. **Neural Bayes Estimator:** Train a neural network to map data to parameter estimates.

Summary of Each Scalable Approach

Approach	**Source of Speedup**	**Tuning Parameters**
Full GP	None	None
Low-rank	Fixed covariance structure: $\mathbf{C}_\Theta = \boldsymbol{\Phi}\boldsymbol{\Phi}^\top + \tau^2\mathbf{I}$	Basis specification for $\boldsymbol{\Phi}$
Covariance Tapering	Taper C_Θ with compact support	Tapering function and radius
Vecchia	Sparse Cholesky factor: $\mathbf{L}_\Theta\mathbf{L}_\Theta^\top = \mathbf{Q}_\Theta = \mathbf{C}_\Theta^{-1}$	# of neighbors and location ordering
Neural Bayes	GPU-accelerated evaluation of neural net	Neural net architecture and number K of training samples
SPQR	Neural net surrogate of Vecchia likelihood	NN architecture, sample size, number of basis functions

Table: Summary of scalable spatial methods, their computational strategies, and tuning parameters.

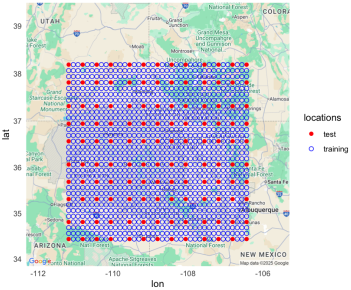
Simulation Study

Simulation Study Design: Emulate ‘skin’ surface temperature from the North American Land Data Assimilation System (NLDAS) in the Four Corners region.

- Locations: 1079 training and 130 test.
- Times: 54 years
- Dependence parameters: $\alpha \in \{0.3, 0.7\}$
- Covariance parameters: $\nu = 1/2$, $\phi = 0.2$
- Replicates: 100 samples

Validation and Performance Metrics:

- Interval scores and coverage for dependence parameter α
- CRPS and tail-weighted CRPS with multiple weighting functions.
- Model-fitting walltimes



Training and test locations

Simulated Data

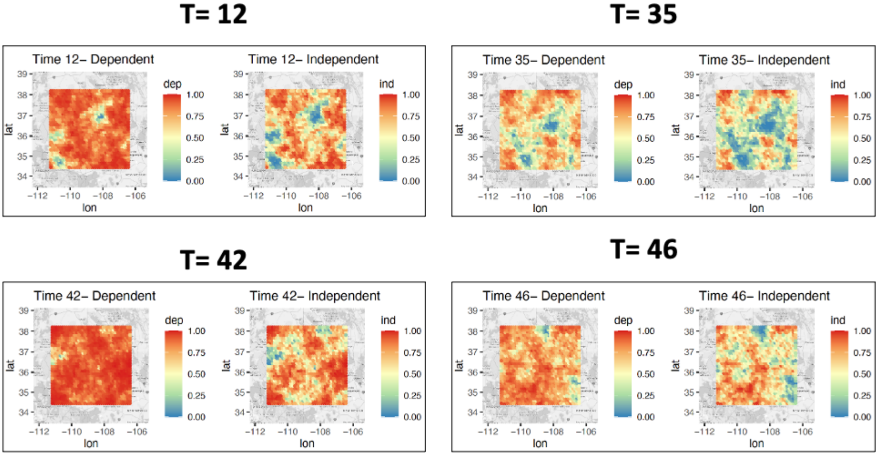


Figure: Simulated snapshots at four time points highlighting differences between the asymptotically dependent and independent classes.

Results: Performance under Dependence ($\alpha = 0.7$)

Takeaways:

- **Accuracy:** Vecchia Approximations and SPQR.
- **Uncertainty Quantification:** Tapering and NBE
- **Speed:** SPQR and NBE

Method	Details	IS	Coverage	CRPS	TWCRPS 1	TWCRPS 2	TWCRPS 3	Walltime (min)
GP	–	0.011	0.650	**0.0451**	0.0473	0.0055	0.0221	138.00
Low Rank A	rank 100	0.034	0.510	0.0794	0.0794	0.0098	0.0384	66.00
Low Rank A	rank 500	0.031	0.580	0.0724	0.0724	0.0087	0.0381	66.00
Low Rank B	rank 100	0.030	0.590	0.0791	0.0791	0.0098	0.0353	66.00
Low Rank B	rank 500	0.035	0.650	0.0715	0.0715	0.0088	0.0348	66.00
Tapering	53% sparse	**0.002**	0.820	0.0473	0.0539	0.0055	0.0227	126.00
Tapering	90% sparse	**0.002**	0.866	0.0504	0.0633	0.0057	0.0245	132.00
Vecchia	NN=5	0.012	0.610	0.0455	0.0477	0.0055	0.0221	84.00
Vecchia	NN=10	0.012	0.680	0.0453	0.0475	0.0055	0.0221	114.00
Vecchia	NN=20	0.013	0.560	0.0452	**0.0474**	**0.0054**	0.0220	252.00
SPQR	NN=10	0.019	0.420	0.0458	0.0480	0.0055	0.0223	11.00
SPQR	NN=20	0.011	0.490	0.0460	0.0489	**0.0054**	**0.0215**	13.00
NBE	–	0.008	**0.920**	–	–	–	–	**0.02**

Amortization time: SPQR requires $1-2$ hours (depending on number of neighbors) and can be used for inference on *any dataset on the same spatial field*.
Neural Bayes requires ~ 48 hours, which can be used for inference on *any dataset*.

Results: Performance under Independence ($\alpha = 0.3$)

Takeaways:

- **Accuracy:** Vecchia Approximations and SPQR.
- **Uncertainty Quantification:** NBE
- **Speed:** SPQR and NBE

Method	Details	IS	Coverage	CRPS	TWCRPS 1	TWCRPS 2	TWCRPS 3	Walltime (min)
GP	–	0.006	0.660	0.0544	0.0568	0.0069	0.0270	138.00
Low Rank A	rank 100	0.282	0.000	0.0952	0.0952	0.0120	0.0464	66.00
Low Rank A	rank 500	0.268	0.000	0.0856	0.0870	0.0108	0.0464	66.00
Low Rank B	rank 100	0.279	0.000	0.0946	0.0946	0.0117	0.0430	66.00
Low Rank B	rank 500	0.266	0.020	0.0856	0.0856	0.0108	0.0424	66.00
Tapering	53% sparse	0.021	0.082	0.0553	0.0621	0.0064	0.0270	138.00
Tapering	90% sparse	0.019	0.140	0.0581	0.0717	**0.0063**	0.0287	132.00
Vecchia	NN=5	**0.004**	0.770	0.0544	0.0578	0.0067	0.0268	84.00
Vecchia	NN=10	0.006	0.710	0.0540	0.0573	0.0066	**0.0267**	114.00
Vecchia	NN=20	0.005	0.720	**0.0539**	**0.0570**	0.0067	**0.0267**	282.00
SPQR	NN=10	0.023	0.410	0.0546	0.0576	0.0066	0.0269	11.00
SPQR	NN=20	0.020	0.450	0.0549	0.0589	0.0065	0.0271	13.00
NBE	–	0.009	**1.000**	–	–	–	–	**0.02**

Discussion

Summary:

- Comparison of scalable approaches for modeling block maxima data accounting for transitions between dependence classes.
- Results:
  - Vecchia approximations and SPQR offer a balance of speed, accuracy, and UQ across dependence regimes.
  - Tapering approaches perform well under asymptotic dependence, but struggles in independence.
  - Low-rank methods underperform across the board.
  - NBE is fast with excellent coverage, but lacks full scoring metrics.

Conclusion and further work

- We build **likelihood-free estimators for spatial (extremal) models**;
- We showcase massive gains in computational and statistical efficiency when using our approach to inference;
- An R interface to the Julia package, **NeuralEstimators** (Sainsbury-Dale, 2024), is available online[2] with censored inference also illustrated[3];
- Not just for spatial data, see André et al. (2025);
- Big comparative study incoming;
- Probably some full posterior inference too!

[2] https://msainsburydale.github.io/NeuralEstimators.jl/dev/

[3] https://github.com/Jbrich95/CensoredNeuralEstimators

André, L. M., Wadsworth, J. L., and Huser, R. (2025). Neural Bayes inference for complex bivariate extremal dependence models. *arXiv:2503.23156*.

References I

André, L. M., Wadsworth, J. L., and Huser, R. (2025). Neural bayes inference for complex bivariate extremal dependence models. *arXiv preprint arXiv:2503.23156*.

Cressie, N., Sainsbury-Dale, M., and Zammit-Mangion, A. (2022). Basis-function models in spatial statistics. *Annual Review of Statistics and Its Application*, 9(1):373–400.

Furrer, R., Genton, M. G., and Nychka, D. (2006). Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*, 15(3):502–523.

Haas, T. C. (1990). Kriging and automated variogram modeling within a moving window. *Atmospheric Environment. Part A. General Topics*, 24(7):1759–1769.

Heaton, M. J., Datta, A., Finley, A. O., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R. B., Hammerling, D., Katzfuss, M., et al. (2019). A case study competition among methods for analyzing large spatial data. *Journal of Agricultural, Biological and Environmental Statistics*, 24:398–425.

Huser, R. and Wadsworth, J. L. (2019). Modeling spatial processes with unknown extremal dependence class. *Journal of the American Statistical Association*, 114(525):434–444.

Katzfuss, M., Guinness, J., Gong, W., and Zilber, D. (2020). Vecchia approximations of gaussian-process predictions. *Journal of Agricultural, Biological and Environmental Statistics*, 25:383–414.

Lenzi, A., Bessac, J., Rudi, J., and Stein, M. L. (2023). Neural networks for parameter estimation in intractable models. *Computational Statistics & Data Analysis*, 185:107762.

References II

Majumder, R., Reich, B. J., and Shaby, B. A. (2024). Modeling extremal streamflow using deep learning approximations and a flexible spatial process. *The Annals of Applied Statistics*, 18(2):1519–1542.

Richards, J., Sainsbury-Dale, M., Zammit-Mangion, A., and Huser, R. (2024). Neural bayes estimators for censored inference with peaks-over-threshold models. *Journal of Machine Learning Research*, 25(390):1–49.

Sainsbury-Dale, M. (2024). *NeuralEstimators: Likelihood-Free Parameter Estimation using Neural Networks*. R package version 0.1-2.

Sainsbury-Dale, M., Zammit-Mangion, A., Cressie, N., and Huser, R. (2025a). Neural parameter estimation with incomplete data. *arXiv preprint arXiv:2501.04330*.

Sainsbury-Dale, M., Zammit-Mangion, A., and Huser, R. (2024). Likelihood-free parameter estimation with neural bayes estimators. *The American Statistician*, 78(1):1–14.

Sainsbury-Dale, M., Zammit-Mangion, A., Richards, J., and Huser, R. (2025b). Neural bayes estimators for irregular spatial data using graph neural networks. *Journal of Computational and Graphical Statistics*, 34(3):1153–1168.

Vecchia, A. V. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):297–312.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32:4–24.

References III

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. (2017). Deep sets. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zhang, L., Shaby, B. A., and Wadsworth, J. L. (2022). Hierarchical transformed scale mixtures for flexible modeling of spatial extremes on datasets with many locations. *Journal of the American Statistical Association*, 117(539):1357–1369.

Fin.

SCAN ME

Scan for full details of my research.