Neural Bayes estimators for likelihood-free and amortised inference for spatial extremes

Jordan Richards[1] Matthew Sainsbury-Dale[2,3]
Andrew Zammit-Mangion[3] Raphaël Huser[2]

[1]School of Mathematics, University of Edinburgh, UK

[2]King Abdullah University of Science and Technology (KAUST), KSA

[3]Centre for Environmental Informatics, National Institute for Applied Statistics Research Australia, University of Wollongong, Australia

Preface

Neural Bayes estimators for likelihood-free
and amortised inference for spatial extremes

Preface

This talk is not for...

Neural Bayes estimators for likelihood-free
and amortised inference for spatial extremes

Preface

This talk is not for...

Neural Bayes estimators for likelihood-free
and amortised inference for spatial extremes

Preface

This talk is not for...

Neural Bayes estimators for likelihood-free
and amortised inference for spatial extremes

Preface

This talk is not for...

Neural Bayes estimators for likelihood-free
and amortised inference for spatial extremes

Preface

This talk **is** for...

Neural Bayes estimators for likelihood-free
and amortised inference for spatial extremes

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

Traditional likelihood-free inference

- Several approaches to **likelihood-free inference**
- **Approximate Bayesian computation (ABC)** or **Indirect inference**:
  - Simulate from a class of models and optimise model parameters by minimising dissimilarity between replicates and observations.
  - **Sensitive** to the choice of summary statistics used to compare simulated and observed data.
  - **Case-specific**, in the sense that ABC generally involves substantial computation each time it is employed.
- **Neural estimators**:
  - Use neural networks to learn the *optimal* summary statistics;
  - Black box - can be applied in many situations and used to create **amortised estimators**, i.e., not case-specific!
- We will focus on inference for spatial extremal processes but the ideas **can be applied more generally**!

Motivating example: max-stable processes

Max-stable processes (MSPs), which arise as the only possible non-degenerate limit of pointwise maxima of i.i.d random fields, are popular models for spatial extremal dependence.
A MSP with unit Fréchet margins has the construction

$$Z(\mathbf{s}) = \sup_{k \geq 1} R_k W_k(\mathbf{s}),$$

where $\{R_k\}_{k \in \mathbb{N}}$ are points of a Poisson process on $(0, \infty)$ with intensity $r^{-2}\mathrm{d}r$ and $\{W_k(\mathbf{s})\}_{k \in \mathbb{N}}$ are i.i.d. copies of a non-negative stochastic process $W(\cdot)$ satisfying $\mathbb{E}[W(\mathbf{s})] = 1$ for all $\mathbf{s} \in \mathcal{S}$.

Motivating example: max-stable processes

- Number of terms in the likelihood grows faster-than-exponentially
- D-th Bell number: $D = 3 \Rightarrow 5$; $D = 10 \Rightarrow 115975$;
- **Computational tractability** of the likelihood is limited (generally) to $D \leq 12$ (Castruccio et al., 2016);
- A lot of time has been spent on researching efficient likelihood-based inference techniques for MSPs, e.g., via **pairwise likelihoods**;
- Computational issues are compounded by (left) censoring (we will come back to this later...)

Castruccio, S., Huser, R., and Genton, M. G. (2016). High-order composite likelihood inference for max-stable distributions and processes. *JCGS* 25.4: 1212-1229.

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 $\Omega(\cdot)$.
  - 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 $L(\boldsymbol{\theta}, \hat{\theta}(\mathbf{Z}))$.
- We use a **neural estimator** that targets the Bayes estimator.

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 the sample space, $\mathcal{S}$, and the parameter space, Θ, with respect to the prior, $\Omega(\cdot)$;

$$r_{\Omega}(\widehat{\boldsymbol{\theta}}(\cdot)) = \int_{\Theta} \left[\int_{\mathcal{S}} L(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}(\mathbf{z})) f(\mathbf{z} \mid \boldsymbol{\theta}) \mathrm{d}\mathbf{z} \right] \mathrm{d}\Omega(\boldsymbol{\theta}),$$

  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 $\Omega(\cdot)$.

Neural Bayes estimators

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

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

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

  where $\boldsymbol{\theta}^{(k)}$, $k = 1, \ldots, K$, is sampled from the prior $\Omega(\cdot)$ 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 $\widehat{\boldsymbol{\theta}}_{\gamma^*}(\cdot)$ approximately inherits the attractive properties of Bayes estimators (e.g., consistency, asymptotic efficiency). See Sainsbury-Dale et al. (2023b).
- The loss function specifies the Bayes estimator and, hence, the neural Bayes estimator.
  - Under the absolute-error loss, the neural Bayes estimator approximates the posterior median.
  - Under the squared-error loss, the neural Bayes estimator approximates the posterior expectation.
  - Under the tilted loss, $(\hat{\theta} - \theta)(\mathbb{I}(\hat{\theta} - q))$, the neural Bayes estimator approximates the posterior q-quantile. See, e.g., Richards et al. (2023).
  - etc.

Sainsbury-Dale, M., Zammit-Mangion, A., & Huser, R. (2023). Likelihood-free parameter estimation with neural Bayes estimators. *The American Statistician*, (In Press), 1-23.

Richards, J., Alotaibi, N., Cisneros, D., Gong, Y., Guerrero, M. B., Redondo, P., & Shao, X. (2023a). Modern extreme value statistics for Utopian extremes *arXiv:2311.11054*

Uncertainty Quantification

Performing principled, fast uncertainty quantification?

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

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

  is the vector of marginal posterior qth-quantiles (Sainsbury-Dale et al., 2023a).

- Therefore, one may approximate a set of marginal posterior quantiles by training a neural Bayes estimator under the loss (2). **No bootstrap!**
- When approximating multiple quantiles (e.g., to construct credible intervals), the neural-network architecture can be designed to prevent quantile crossing.

Sainsbury-Dale, M., **Richards, J.**, Zammit-Mangion, A., & Huser, R. (2023). Neural Bayes estimators for irregular spatial data using graph neural networks. arXiv:2310.02600

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 $\Omega(\cdot)$, 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),

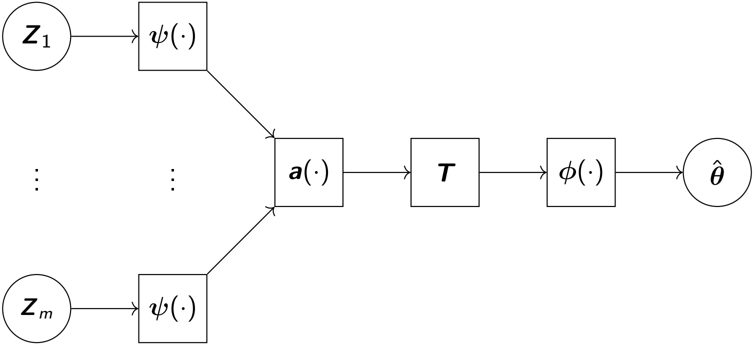
$$\hat{\boldsymbol{\theta}}(\boldsymbol{Z}) = \boldsymbol{\phi}(\boldsymbol{a}(\{\boldsymbol{\psi}(\boldsymbol{Z}_i)\}_{i=1,\ldots,m})),$$

  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.

- Universality of DeepSets (Wagstaff et al., 2022) means that we can approximate a large class of Bayes estimators arbitrarily well.

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.
- Graph neural networks for irregularly-observed spatial data. Agnostic to number and configuration of sampling locations (Sainsbury-Dale et al., 2023a).
- Extensions: LSTMs, CNN-LSTMs, spherical CNNs...
- What do we do if our data are **censored**?

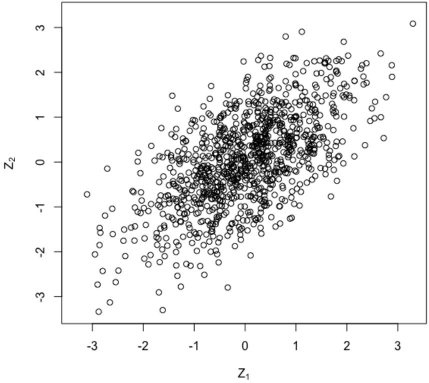
Sainsbury-Dale, M., **Richards, J.**, Zammit-Mangion, A., & Huser, R. (2023). Neural Bayes estimators for irregular spatial data using graph neural networks. arXiv:2310.02600

Background: peaks-over-threshold models

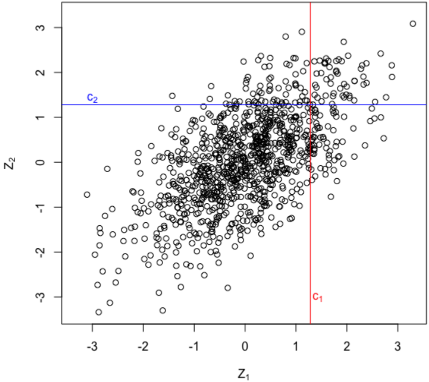
- When doing inference for extremal dependence, we might actually **choose to treat our data as censored!**
- Likelihood estimators for spatial extremal dependence models are typically highly biased if spatial extreme events include marginally non-extreme values (Huser et al., 2016);
- Can be mitigated 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.

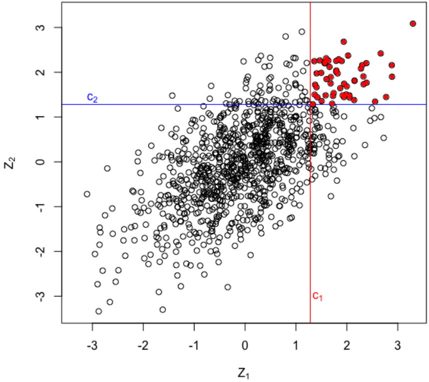
Background: peaks-over-threshold models



Background: peaks-over-threshold models



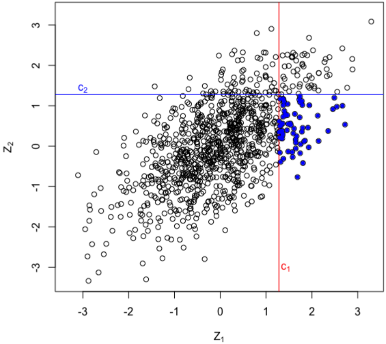
Background: peaks-over-threshold models



Both components extreme ⇒ **both fully observed**.

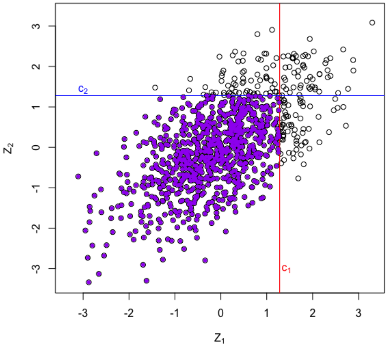
Likelihood contribution of (Z_1, Z_2): $f(z_1, z_2)$

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.

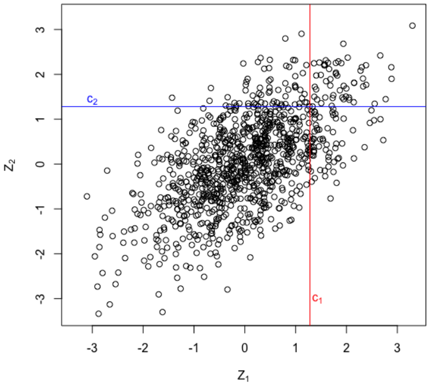
Defining censored inputs

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 $\mathbf{Z}$.

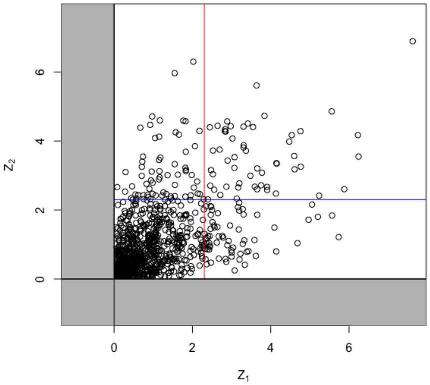
NBE input specification

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



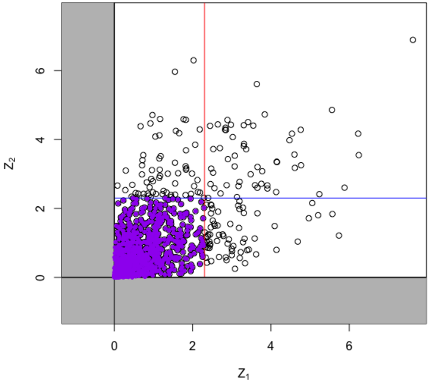
NBE input specification

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 $\mathbf{Z}$).



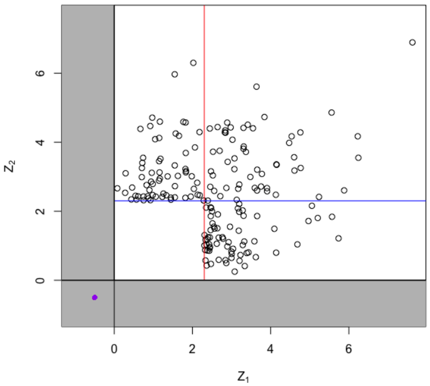
NBE input specification (cont.)

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



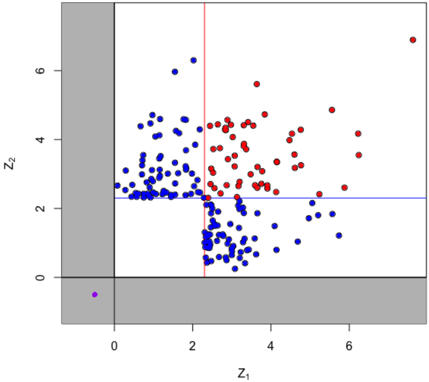
NBE input specification (cont.)

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



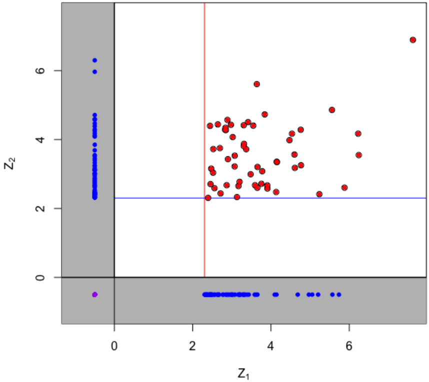
NBE input specification (cont.)

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.)

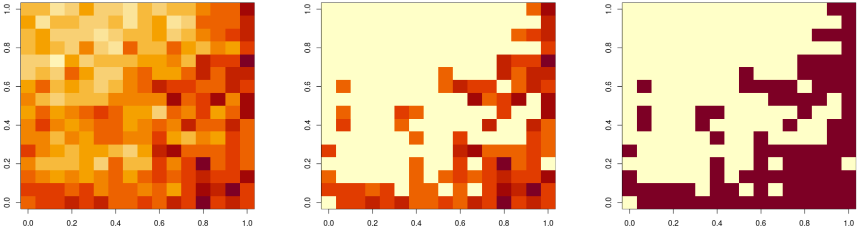
Information about **extreme** components is retained and will continue to contribute to estimation of $\boldsymbol{\theta}$.



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),
- HW process (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.

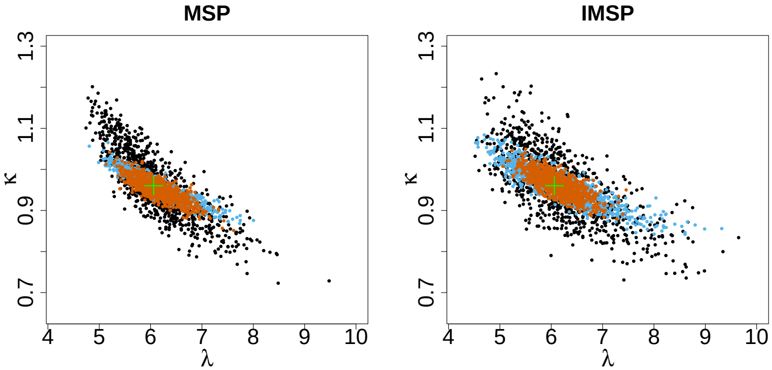
Simulation study 1: results

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 ≈ 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[1] 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.

[1]Constraint alleviated by Sainsbury-Dale et al. (2023a)

Simulation study 2: outline

- Simulate $m = 200$ replicates of a HW process on a 16×16 grid in $[0, 16] \times [0, 16]$;
- Model has three parameters with priors $\lambda \sim \text{Unif}(0.2, 10)$, $\kappa \sim \text{Unif}(0.5, 2)$ and $\delta \sim \text{Unif}(0, 1)$;
- **For a test censoring level** $\tau^* = 0.919$, we compare two NBEs; one trained with τ fixed at $\tau = \tau^*$ and one with τ randomly drawn from a $\text{Unif}(0.85, 0.95)$ for each set of replicates used for training;

Simulation study 2: results

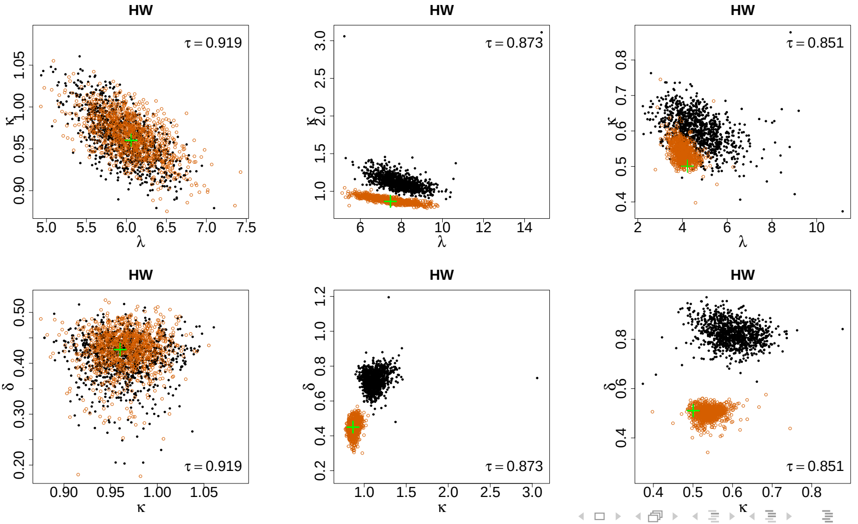
Marginal test risk (s.d.) evaluated on 1000 **test parameter sets with censoring level** τ^*.

τ	λ	κ	δ
random	**2.62 (0.07)**	**2.13 (0.05)**	**2.98 (0.09)**
fixed	2.75 (0.06)	2.41 (0.06)	3.25 (0.10)

- We can train an estimator for a general τ.
- Randomising τ during training improves the estimator performance.
- **Implication**: a new user will not need to retrain an estimator if they want to use a different censoring level.

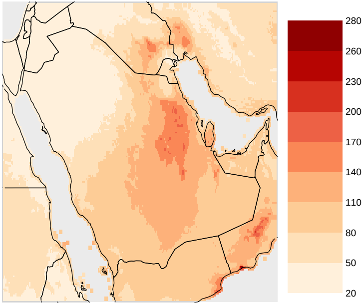
Simulation study 2: joint distribution

Different τ: (left) 0.919, (centre) 0.873, (right) 0.851.



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. ($\mu g/m^3$) 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.)

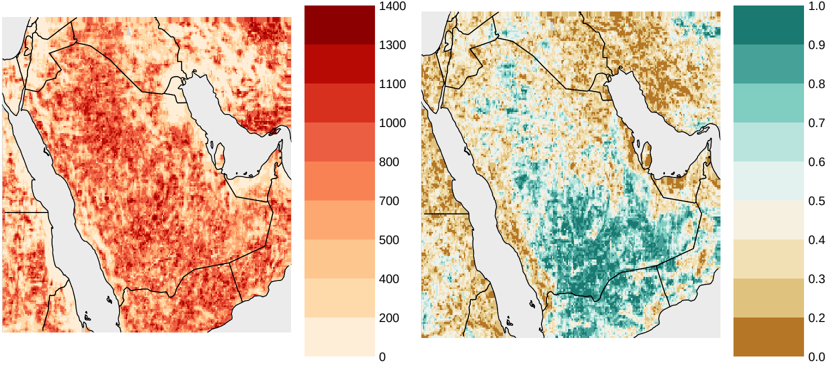
- Data are arranged on a 242×182 regular grid; monthly, 1998–2020.
- Fit local anisotropic HW processes with $\tau = 0.9$ (five params.);
- To all possible subsets of data on $G \times G$ grids for smoothing level $G \in \{4, 8, 16, 24, 32\}$. This is over 130,000 fits!
- Once an estimator is trained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

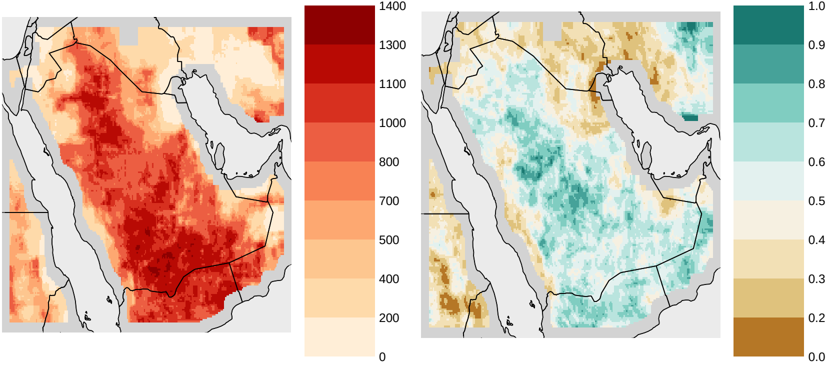
Each pixel is a single model fit.



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

[2] as far as we know.

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[2]!**

[2] as far as we know.

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[2]!**
- And only five estimators have been trained (one for each G).

[2] as far as we know.

Conclusion and further work

- We build **likelihood-free estimators for peaks-over-threshold spatial extremal dependence models**;
- We showcase massive gains in computational and statistical efficiency when using our approach to inference;
- An R interface to our Julia package, **NeuralEstimators**, is available online[3] with censored inference also illustrated[4];
- Recent additions using NBEs for irregular spatial data (Sainsbury-Dale et al., 2023a).

[3] https://github.com/msainsburydale/NeuralEstimators

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

Sainsbury-Dale, M., Richards, J., Zammit-Mangion, A., & Huser, R. (2023a). Neural bayes estimators for irregular spatial data using graph neural networks. *arXiv:2310.02600*

References

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.

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.

Richards, J., Sainsbury-Dale, M., Zammit-Mangion, A., and Huser, R. (2023). Likelihood-free neural bayes estimators for censored inference with peaks-over-threshold models. *arXiv:2306.15642*.

Sainsbury-Dale, M., Richards, J., Zammit-Mangion, A., and Huser, R. (2023a). Neural bayes estimators for irregular spatial data using graph neural networks. *arXiv preprint arXiv:2310.02600*.

Sainsbury-Dale, M., Zammit-Mangion, A., and Huser, R. (2023b). Likelihood-free parameter estimation with neural Bayes estimators. *The American Statistician*. in press (arXiv version available: arXiv:2208.12942).

Wagstaff, E., Fuchs, F. B., Engelcke, M., Osborne, M., and Posner, I. (2022). Universal approximation of functions on sets. *Journal of Machine Learning Research*, 23:1–56.

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.

Fin.

SCAN ME

Scan for full details of my research.