Estimating a Directed Tree for Extremes

Ngoc Mai Tran†

Department of Mathematics, University of Texas at Austin, Speedway 2515 Stop C1200, Austin TX 78712, USA, email: ntran@math.utexas.edu

Johannes Buck‡ and Claudia Klüppelberg

Department of Mathematics, Technical University of Munich, 85748 Garching, Boltzmannstr. 3, Germany, emails: j.buck@tum.de, cklu@cit.tum.de

Summary. The Extremal River Problem has emerged as a flagship problem for causal discovery in extreme values of a network. The task is to recover a river network from only extreme flow measured at a set V of stations, without any information on the stations' locations. We present `QTree`, a new simple and efficient algorithm to solve the Extremal River Problem that performs very well compared to existing methods on hydrology data and in simulations. `QTree` returns a root-directed tree and achieves almost perfect recovery on the Upper Danube network data, the existing benchmark data set, as well as on new data from the Lower Colorado River network in Texas. It can handle missing data, has an automated parameter tuning procedure, and runs in time $O(n|V|^2)$, where n is the number of observations and $|V|$ the number of nodes in the graph. Furthermore, we prove that the `QTree` estimator is consistent under a Bayesian network model for extreme values with noise. We also assess the small sample behaviour of `QTree` through simulations and detail the strengths and possible limitations of `QTree`.

1. Introduction

Causal inference from extremes aims to discover cause and effect relations between large observed values of random variables. To understand causality of high risk variables is much needed as rare events like environmental or financial risks are often cascading through a network. Pollutants can propagate through an unseen underground waterway, causing extreme measurements at multiple locations (Leigh et al., 2019). Credit markets might fail due to some endogenous systemic risk propagation (Rochet and Tirole, 1996). However, it is not immediately obvious how to extend the past decades of work on causal inference (Bollen, 1989; Drton and Maathuis, 2017; Lauritzen, 1996; Maathuis et al., 2019; Pearl, 2009; Spirtes et al., 2000) for Gaussian and discrete distributions to extreme values. Since the focus is on maxima rather than averages, correlations or other bivariate measures of dependence in the center of the distribution are replaced by *extreme dependence measures* (Coles et al., 1999; Engelke and Volgushev, 2020; Larsson and Resnick, 2012; Sibuya, 1960), which are often difficult to estimate from limited data.

These extreme dependence measures are derived through asymptotic theory from generalized extreme value distributions—modelling sample extremes—or generalized Pareto

†Supported by NSF Grant DMS-2113468 and NSF IFML 2019844 Award.
‡Supported by the Hanns Seidel Foundation

distributions—modelling excesses over high thresholds. Textbook treatments can be found in (Beirlant et al., 2004; Coles et al., 2001; de Haan and Ferreira, 2007; Resnick, 1987, 2007). A very readable review paper is Davison and Huser (2015). The development of graphical models for extremes follows these two approaches. Max-linear causal graphical models or max-linear Bayesian networks are motivated by extreme value distributions, which are max-stable (closed with respect to taking maxima). Introduced in Gissibl and Klüppelberg (2018), they are defined via a max-linear recursively defined structural equation models on a directed acyclic graph (see Pearl et al. (2009)). Each node represents a positive random variable defined as a weighted maximum of its parent variables and an independent innovation. The multivariate distribution of a max-linear vector is not restricted to a Fréchet distribution; indeed the independent innovations can have arbitrary continuous distributions. The emphasis of the model is on its structure given by a directed graph. Nonparametric statistical inference aims at identifying the directed graphical structure regardless of the node distributions.

Engelke and Hitz (2020), on the other hand, define a new extreme conditional dependence concept for multivariate Pareto distributions and use this concept to define extreme graphical models similarly as the classical concept for densities. The multivariate distribution determines the model and has to be specified for statistical inference. Here the multivariate Hüsler-Reiss distribution plays a prominent role; see Asenova et al. (2021); Asenova and Segers (2022); Engelke and Volgushev (2020); Engelke et al. (2022); Hu et al. (2022); Rötter et al. (2022).

Motivated by extreme value theory, it is not surprising that max-linear Bayesian networks have been mostly investigated for heavy-tailed innovations: (Einmahl et al., 2018, Section 3.3) consider tail dependence functions for i.i.d. Fréchet innovations. Gissibl et al. (2018) investigate tail dependence for i.i.d. regularly varying innovations, and Klüppelberg and Krali (2021) the scaling properties of the same model. Asenova et al. (2021) and Segers (2020) investigate regularly varying Markov trees, and more recently, Asenova and Segers (2022) investigate a new max-linear graphical model on trees of transitive turnaments.

1.1. The Extremal River Problem

The relevance of extremal graphical models for multivariate distributions has been validated on several data sets, prominently on the Upper Danube river network. The goal is to recover a river network from *only extreme flow* measured at a set V of stations, *without* any information on the stations' location. We refer to it as the *Extremal River Problem*. Here, the true river network serves as the 'gold standard', allowing one to verify the performance of the proposed estimator. Success in solving the Extreme River Problem can translate to new solutions to the *contaminant tracing* challenge in hydrology (Leigh et al., 2019; McGrane, 2016; Rodriguez-Perez et al., 2020; Ver Hoef and Peterson, 2010; Ver Hoef et al., 2006; Wolf et al., 2012). There, one needs an inexpensive method to trace pollutants or chemical constituents transported by a complex and unknown underground waterway that is prohibitive to model or survey with traditional fluid mechanics methods (Anderson et al., 2015). Recent advances point towards an imminent data explosion (Bartos et al., 2018; Mao et al., 2019), where pollutants

exceeding certain thresholds can be detected via a sensor network. Thus, contaminant tracing with sensors data is a version of the Extremal River Problem without the gold standard, where the network is truly unknown.

The Extremal River Problem with test data given by river discharges of the Upper Danube river network has proven to be challenging and very stimulating for extreme value theory, with each paper taking a different technique. The data have been pre-processed in Asadi et al. (2015) and are available in the `R` package `graphicalExtremes` (Engelke et al., 2019).

The preprocessed data have been analysed in a number of publications with focus on modelling extreme dependence: flow- and spatial dependence (Asadi et al., 2015) and undirected graphical models for extremes (Engelke and Hitz, 2020; Engelke et al., 2022; Hu et al., 2022; Gong et al., 2022; Rötter et al., 2022). In a first paper, Engelke and Hitz (2020) returned a highly accurate but *undirected* graph, followed by publications using new models and applying different methods for reconstructing the undirected graph.

Our focus is on causality in extremes, modelled by a directed tree; hence, a large value at node j causes a large value at node i, whenever there is an edge from j to i. Similar problems have also been considered modelling causal extreme dependence by expected quantile scores in (Mhalla et al., 2020) and by causal dependence coefficients in (Gnecco et al., 2021). Gnecco et al. (2021) correctly recovered the causal order of 12 nodes out of 31, but did not learn the entire river network, while Mhalla et al. (2020) focused on flow-connections and did well at detecting nodes connected by a directed path; see Figure 7 in Mhalla et al. (2020). These two publications have slightly different notions of causality; we discuss this in Section 4.

1.2. Main contributions and structure of the paper

The novelty of our paper lies in several directions.

(a) We suggest a new algorithm `QTree` to recover causality in extremes, where causality is modelled by a directed tree. The advantage of the proposed max-linear Bayesian tree is a structural model, which does not require to specify multivariate distributions. Moreover, no normalization of the data to standard Fréchet or Gumbel is needed.

(b) The `QTree` algorithm estimates a pair-wise score matrix W, and applies Chu–Liu/-Edmonds' algorithm to output a root-directed spanning tree of optimum score. The algorithm is simple, using properties of a max-linear Bayesian tree, and has a stabilizing subsampling procedures that is based on bootstrap aggregation.

(c) We prove strong consistency of the estimated trees in an extreme value setting with possible noise as the sample size tends to infinity. This proof is based on a new variational argument to account for noise in the data.

(d) We analyse three new data sets from the Lower Colorado river network in Texas. We also show by a simulation study that `QTree` is robust with respect to different dependence structures (given by edge-weights) and different node distributions entailed from different innovations distributions.

`QTree` performs extremely well on real-world data sets. Algorithm 2 achieves almost perfect recovery of the Upper Danube network. In addition to the Upper Danube, we

further test `QTree` on three sectors of the Lower Colorado river network in Texas. These are much more challenging data sets. The Colorado river network suffers from severe drought, extreme flooding, and sensors failure, with up to 36.9% missing data (the Upper Danube has none). These challenges make recovering the Lower Colorado network much closer to the trace contaminant challenge. Remarkably, on all three sectors of the Lower Colorado, `QTree` Algorithm 2 also achieves almost perfect recovery (cf. Section 4).

Beyond hydrology, `QTree` can be applied to cause and effect detection in every high risk problem assuming that the network is a root-directed tree. At a high level, `QTree` aims to fit a *max-linear Bayesian tree* to the data. Max-linear Bayesian networks have recently emerged as a suitable directed graphical model for causality in extremes (Améndola et al., 2022; Buck and Klüppelberg, 2021; Gissibl, 2018; Gissibl and Klüppelberg, 2018), however, existing methods for learning them aim to learn the model parameters and thus are highly sensitive to model misspecifications; see Buck and Klüppelberg (2021); Gissibl (2018); Gissibl et al. (2021); Gissibl et al. (2018); Klüppelberg and Krali (2021); Klüppelberg and Lauritzen (2020). In particular, they do not perform well on the Upper Danube data set. In contrast, `QTree` relies on *qualitative* aspects of the max-linear Bayesian network model to score each potential edge independently, and then applies Chu–Liu/Edmonds' algorithm to return an optimal root-directed spanning tree; see e.g. Gabow et al. (1986), Section 3. We detail the algorithm and the intuition behind it in Section 2. Note that `QTree` heavily relies on the assumption that there are sufficiently many extreme observations, and that the signal has heavier tail than the noise. Assuming that the data come from a noisy max-linear Bayesian network with appropriate signal-to-noise ratio, we prove that the tree output by `QTree` is strongly consistent (cf. Theorem 1).

`QTree` is very flexible, has only two tuning parameters, and is very efficient. It runs in time $O(n|V|^2)$, where n is the number of observations and $|V|$ is the number of nodes. `QTree` maximizes the information available from missing data since at each step it only utilizes the data projected onto two coordinates. `QTree` is implemented as a plug-and-play package in `Python` (Tran, 2021) at

`https://github.com/princengoc/qtree`

which includes all data and codes to produce the results and figures in this paper.

Our paper is organized as follows. We introduce `QTree` (Algorithm 1) and `auto-tuned QTree` (Algorithm 2) in Section 2 and discuss its intuition supported by preliminary simulation results. In Section 3, we present the data sets, discuss their specific challenges and describe the data preprocessing steps. In Section 4, we present the estimation results of `QTree` and analyze the performance of the automated parameter selection. Here we also compare different algorithms in the literature with ours. In Section 5, we test the limits of `QTree` by a small simulation study. Section 6 concludes with a summary. The Supplementary Material includes the proof of the Consistency Theorem (Theorem 1).

Notations. Estimators are compared based on standard metrics in causal inference (Zheng et al., 2018): normalized structural Hamming distance (nSHD), false dicovery rate (FDR), false positive rate (FPR), and true positive rate (TPR). All of these metrics lie between 0 and 1. We recall their definitions here. Let $\mathcal{G}$ be the true graph and $\hat{\mathcal{G}}$ an estimated graph. The *structural Hamming distance* $\mathrm{SHD}(\mathcal{G}, \hat{\mathcal{G}})$ between $\mathcal{G}$ and $\hat{\mathcal{G}}$ is the minimum number of edge additions, deletions and reversals to obtain $\mathcal{G}$ from $\hat{\mathcal{G}}$. Denote

$E(\mathcal{G})$ and $E(\hat{\mathcal{G}})$ the set of edges in $\mathcal{G}$ and $\hat{\mathcal{G}}$, respectively. Note that $|E(\hat{\mathcal{G}}) \setminus E(\mathcal{G})|$ is the number of edges in $\hat{\mathcal{G}}$ that are not in $\mathcal{G}$, while $|E(\hat{\mathcal{G}}) \cap E(\mathcal{G})|$ is the number of correctly estimated edges. We then have

$$\begin{aligned} \mathrm{nSHD}(\hat{\mathcal{G}}, \mathcal{G}) &:= \frac{\mathrm{SHD}(\hat{\mathcal{G}}, \mathcal{G})}{|E(\hat{\mathcal{G}})| + |E(\mathcal{G})|}, & \mathrm{FDR}(\hat{\mathcal{G}}, \mathcal{G}) &:= \frac{|E(\hat{\mathcal{G}}) \setminus E(\mathcal{G})|}{|E(\hat{\mathcal{G}})|}, \\ \mathrm{FPR}(\hat{\mathcal{G}}, \mathcal{G}) &:= \frac{|E(\hat{\mathcal{G}}) \setminus E(\mathcal{G})|}{|V| \times (|V| - 1) - |E(\mathcal{G})|}, & \mathrm{TPR}(\hat{\mathcal{G}}, \mathcal{G}) &:= \frac{|E(\hat{\mathcal{G}}) \cap E(\mathcal{G})|}{|E(\mathcal{G})|}. \end{aligned} \tag{1}$$

The performance of an algorithm is better the smaller the first three metrics are and the larger TPR is. We shall use this throughout Section 4.

2. The algorithm

2.1. The data generation model

Throughout we assume data on a root-directed spanning tree $\mathcal{T}$ on V nodes. That is, each node $i \in V$ except the root r has exactly one child, the root r has none, and there is a path from every node $i \neq r$ to r. In the Extremal River Problem, our goal is to recover the unknown $\mathcal{T}$ from extreme discharges X_i at nodes $i \in V$. Our starting point is the max-linear Bayesian network (Gissibl and Klüppelberg, 2018), a model for risk propagation in a directed acyclic graph. When the graph is a tree $\mathcal{T}$, then the model is defined as

$$X_i = \bigvee_{j : j \to i \in \mathcal{T}} c_{ij} X_j \vee Z_i, \quad c_{ij}, Z_i > 0, \quad i \in V. \tag{2}$$

Here the Z_i, called innovations, are independent with support $\mathbb{R}_+$ and have atom-free distributions. The model says that each edge $j \to i$ in $\mathcal{T}$ has a weight $c_{ij} > 0$, interpreted as some measure of the flow rate from j to i, and an extreme discharge at i is either the result of an unknown external input Z_i (e.g. heavy rainfall), or it is the maximum of weighted discharges Z_j from an ancestral node j of i.

Here, the root-directed tree $\mathcal{T}$ describes the causal structure in the data. The Extremal River Problem aims at finding this tree from extreme observations only. Indeed, in Tran (2022), Section 2.1 it is shown that for the Upper Danube data also a naive algorithm based on the pairwise correlation matrix as score matrix performs well, whereas for the Lower Colorado data it returns a less precise tree. So this is an example, where the extremes contain more causal information than the average observations.

`QTree` can, however, also solve a slightly more general problem. Assume that the tree structure is only in the extremes, whereas "average" data follow a different model. It can also happen that only data from certain nodes follow a heavy-tailed distribution (able to model extreme events, while other nodes are negligible from an extreme value point of view; see e.g. Embrechts et al. (1997); de Haan and Ferreira (2007); Resnick (1987, 2007)). Then it may well be possible that the causality in the extremes can be modeled by a tree on a subset of nodes.

For numerical stability, we prefer to work with the logarithm of the extreme data. To avoid new symbols, we keep the same notation, so the max-linear Bayesian tree becomes

$$X_i = \bigvee_{j:j\to i\in\mathcal{T}} (c_{ij} + X_j) \vee Z_i, \quad c_{ij}, Z_i \in \mathbb{R}, \quad i \in V. \tag{3}$$

We further assume that data is corrupted with independent noise in each coordinate. The Extremal River Problem thus becomes the following.

> **The Extremal River Problem**. Given n observations $\mathcal{X} = \{x^1+\varepsilon^1, \dots, x^n+\varepsilon^n\}$ in $\mathbb{R}^V$, where the x_i are generated via (3), and the ε_i are independent noise variables in $\mathbb{R}^V$, find $\mathcal{T}$.

We stress that the root-directed tree assumption is *different* from the usual tree in Bayesian networks, where each *child* has at most one parent. Learning the single-parent tree can be done with the message passing algorithm, which recursively identifies the parent of a node through likelihood calculations (Wainwright and Jordan, 2008). This strategy does not work for the root-directed tree, since each child can have multiple parents.

2.2. Intuition of `QTree`

In general, learning Bayesian networks with more than one parent is NP-hard, see (Chickering, 1996). However, learning the max-linear Bayesian network from i.i.d *noise-free observations* is solvable in time $O(|V|^2 n)$ with $O(|V|(\log(|V|))^2)$ observations (cf. Lemma S1 in the Supplementary Material). Here is the intuition.

Fix an edge $j \to i$ and consider the noise-free model (3). If for an observation $x \in \mathbb{R}^V$ the value at j causes that at i, then $x_i = c_{ij}+x_j$. If j does not cause i, then $x_i > c_{ij}+x_j$. Over n independent observations, if the value at j causes the value at i at least twice, then the distribution of $x_i - x_j$ has an *atom* at its left-end point. Repeating this argument shows that if j causes k and k causes i, then one also has $x_i - x_j = c_{ik} + c_{kj}$. That is, if the sample $\mathcal{X}$ is noise-free, the empirical distribution of

$$\mathcal{X}_{ij} := \{x_i - x_j : x \in \mathcal{X}\} \tag{4}$$

has for sufficiently many observations multiple values *at* the minimum of its support *if and only if* $j \rightsquigarrow i$. Thus, with enough observations, one can recover the directed path $j \rightsquigarrow i$, from which $\mathcal{T}$ can be uniquely constructed as it is a root-directed tree.

`QTree` exploits the above intuition and makes it work under the presence of noise. Consider an ordered pair of nodes $(j, i) \in V$. If the noise at i is small relative to the signal at j, one can expect a concentration *near* the minimum of $\mathcal{X}_{ij}$ if and only if $j \rightsquigarrow i$. This is the intuition of `QTree`. While we have no control over the noise, one way to obtain 'strong signals x_j' is to replace (4) by the set

$$\mathcal{X}_{ij}(\alpha) := \{x_i - x_j : x \in \mathcal{X}, x_j > Q_{\mathcal{X}_j}(\alpha)\}, \tag{5}$$

where $Q_{\mathcal{X}_j}(\alpha)$ is the α-th quantile of the empirical distribution of $\mathcal{X}$ in the j-th coordinate. For $\alpha > 0$, this amounts to a transformation of $\mathcal{X}_{ij}$ that amplifies its concentration near the minimum, at the cost of keeping only a fraction of the available observations (cf. Figure 1).

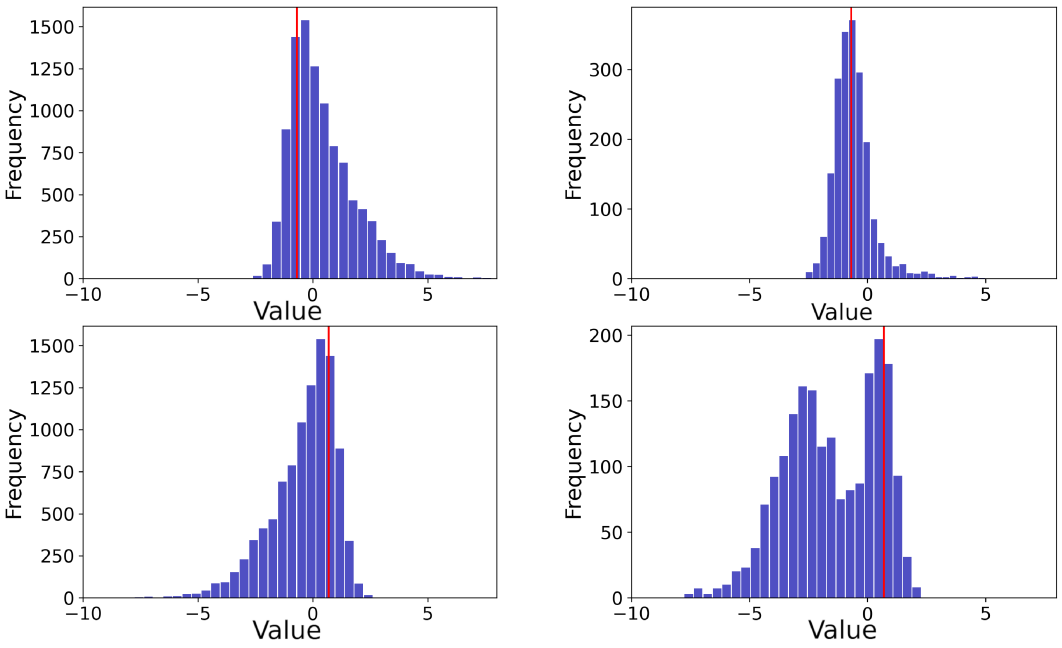

Fig. 1. For the simple graph $1 \to 2$ with $c_{21} = \log(0.5) = -0.69$ and normal centered noise with standard deviation 0.5: *First row:* Histograms of the observations $\mathcal{X}_{21}$ as in (4) (left) and truncated observations $\mathcal{X}_{21}(0.8)$ as in (5). In most cases, for $\mathcal{X}_{21}(0.8)$ depicted in the right-hand figure, a large value for x_2 is realised from a large value of x_1, hence the increasing symmetry around c_{21}. *Second row:* Histograms of the observations $\mathcal{X}_{12}$ and $\mathcal{X}_{12}(0.8)$ corresponding to the reversely directed edge. For $\mathcal{X}_{12}(0.8)$ depicted in the right-hand figure, a large value x_2 is realised either from large x_1 or from a large innovation Z_2, giving the bimodal distribution. The vertical lines show the position of the atom in the noise-free distribution, i.e. $\log(0.5)$ in the first row and $-\log(0.5)$ in the second row.

2.3. The `QTree` *Algorithm*

The `QTree` Algorithm 1 computes independently for each potential edge $j \to i$ a score w_{ij}, seen as a measure of concentration of $\mathcal{X}_{ij}(\alpha)$ near its minimum, then outputs a minimum directed spanning tree of the graph $\mathcal{G}$ with scores $W = (w_{ij})$. The idea is that at each node j, data would show the highest concentration at the true edge among all edges from some parent of j to j. Theorem 1 proves this for the Gumbel-Gaussian noise model; see around (9). The default concentration measure for `QTree` is the empirical *quantile-to-mean gap*

$$w_{ij}(\underline{r}) := \frac{1}{n_{ij}} \left(\mathbb{E}(\mathcal{X}_{ij}(\alpha)) - Q_{\mathcal{X}_{ij}(\alpha)}(\underline{r}) \right)^2, \tag{6}$$

where $\mathbb{E}$ is the empirical mean, Q the empirical quantile, $\underline{r} \in (0, 1)$ is a small quantile level and $n_{ij} = |\mathcal{X}_{ij}(\alpha)|$ is the number of observations in the set $\mathcal{X}_{ij}(\alpha)$ defined in (5). The normalization factor n_{ij} only matters when missing values are unevenly distributed across pairs, such as for the Lower Colorado network (cf. Section 3.2). Then, pairs with fewer observations get a relative penalty in the concentration estimate to account for larger variability in sample quantile estimates due to a small sample size. If no missing values are present, as is the case with the Upper Danube network, then $n_{ij} =$

$n \cdot (1-\alpha)$ for all pairs (i,j) and the algorithm would return the exact same tree $\hat{\mathcal{T}}$ as if the concentration measure was defined without dividing by n_{ij}.

We note that there are other choices for a concentration measure, such as the empirical *lower quantile gap*,

$$w_{ij}(\underline{r},\overline{r}) := \frac{1}{n_{ij}} \left(Q_{\mathcal{X}_{ij}(\alpha)}(\overline{r}) - Q_{\mathcal{X}_{ij}(\alpha)}(\underline{r})\right)^2, \tag{7}$$

where $0 < \underline{r} < \overline{r} < 1$ is a fixed pair of quantile levels. If $\overline{r}$ is small, then $w_{ij}(\underline{r},\overline{r})$ is a local measure of concentration in the lower tail of $\mathcal{X}_{ij}(\alpha)$. Note that, if the number of observations is small, then $\overline{r}$ cannot be too small, so the two empirical concentration measures are in fact rather similar on a real data set. In practice, the lower quantile gap has one more parameter to tune, and thus we choose the quantile-to-mean gap as our default.

Remark 1. Observe that both concentration measures, (6) and (7), are translation invariant. Consequently, considering log data, both measures are invariant to scaling.

Algorithm 1 `QTree` for fixed parameters

Parameters: $\underline{r} \in (0,1)$, $\alpha \in [0,1)$.
Input: data $\mathcal{X} = \{x^1, \ldots, x^n\} \subset \mathbb{R}^V$.
Output: a root-directed spanning tree $\hat{\mathcal{T}}$ on V.

1: **for** $j \to i$, $j, i \in V, j \neq i$ **do**
2: Compute $w_{ij}(\underline{r})$ by (6).
3: Compute $\hat{\mathcal{T}}$:= minimum root-directed spanning tree on the directed graph $(V, \mathcal{G})$ with score matrix $W = (w_{ij}(\underline{r})) \in \mathbb{R}^{V \times V}$ with Chu–Liu/Edmonds' algorithm with variable root.
4: **Return** $\hat{\mathcal{T}}$

Remark 2. Given a score matrix W (equivalently a bidirected graph) and a unique root (the initial node), Chu–Liu/Edmonds' algorithm (see Gabow et al. (1986) and Grötschel et al. (1988), Sections 7.2 and 8.4 for more background) finds a minimum directed spanning tree; i.e., a network of minimum score with $\sum_{j:j\to i\in\hat{\mathcal{T}}} w_{ij}(\underline{r})$ as small as possible. As we want a minimum *root-directed* spanning tree, we simply reverse edge directions. Moreover, we run the algorithm for every possible node as root, and take a tree with minimum score. Finally, provided that all scores are different, the algorithm finds a unique minimum root-directed spanning tree.

2.3.1. Theoretical properties of `QTree`

We prove consistency of Algorithm 1 under natural conditions on the distribution of the innovations. We focus on the structural tree model of a max-linear Bayesian network as introduced in Gissibl et al. (2018) taking i.i.d. innovations with Frechét distribution function $P(Z_i \leq x) = e^{-x^{-\alpha}}, x > 0$, for $\alpha > 0$. Then, using the solution of (2) given in Theorem 2.2 of Gissibl and Klüppelberg (2018), by max-stability (e.g. Embrechts

et al. (1997), Section 3.2), X is multivariate Fréchet distributed with marginals as in Proposition A.2 of Gissibl et al. (2018)):

$$P(X_i \leq x) = \exp\{-(x_i\mu_i)^{-\alpha}\}, \quad x > 0,$$

for $\mu_i = \left(\sum_{j:j\rightsquigarrow i, j=i} {c_{ji}}^{\alpha}\right)^{-1/\alpha}$. Taking logarithms of the X_i is equivalent to taking logarithms of the innovations Z_i with $P(\log(Z_i) \leq x) = \exp\{-e^{-x/\beta}\}, x \in \mathbb{R}$, for $\beta := 1/\alpha > 0$. This results in a Gumbel model

$$P(X_i \leq x) = \exp\{-e^{-(x-\mu_i)/\beta}\}, \quad x \in \mathbb{R}\,.$$

Therefore, for log data, the X_i are Gumbel(β, μ_i) distributed with scale $\beta := 1/\alpha$ and location μ_i.

Instead of taking the logarithmic analog of a *Generalised* Fréchet model as often done in the literature, we prefer instead to add a small independent noise to the max-linear Bayesian tree model (3) with Gumbel$(\beta, 0)$ innovations. This motivates the *noise model*

$$X_i = \Big(\bigvee_{j:j\to i\in\mathcal{T}} (c_{ij} + X_j) \vee Z_i \Big) + \varepsilon_i, \quad c_{ij}, Z_i, \varepsilon_i \in \mathbb{R}, \quad i \in V. \tag{8}$$

with the following innovation-noise distributions:

> **Gumbel-Gaussian noise model.** For $i \in V$, the innovations Z_i are i.i.d. Gumbel$(\beta, 0)$, the noise variables ε_i are i.i.d with symmetric, light-tailed density f_ε satisfying
>
> $$f_\varepsilon(x) \sim e^{-Kx^p} \text{ as } x \to \infty, \tag{9}$$
>
> for some $p > 1$ and $\gamma, K > 0$ and the derivative of f_ε exists in the tail region. Throughout, for two functions a, b, positive in their right tails, we write $a(x) \sim b(x)$ as $x \to \infty$ for $\lim_{x\to\infty} a(x)/b(x) = c$, where $c > 0$ is some arbitrary constant.

Remark 3. The density f_ε in (9) belongs to a special class of light-tailed densities whose convolution tail can be derived asymptotically (Balkema et al., 1993). The family includes the Gaussian ($p = 2$), and though it is strictly more general than the Gaussian, we follow Balkema et al. (1993), and call our noise model Gumbel-Gaussian for ease of reference. Condition (9) guarantees that the upper tail of $\varepsilon_i - \varepsilon_j$ is *lighter* than that of $Z_i - Z_j$ (cf. Lemma S4 in the Supplementary Material).

Theorem 1 below, proved in Section S2 of the Supplementary Material, says that under the Gumbel-Gaussian noise model, both quantile-to-mean and lower quantile gap produce together with Chu-Liu/Edmonds' algorithm strongly consistent estimators for the true root-directed spanning tree $\mathcal{T}$ for appropriate choice of parameters. Simulation results (cf. Figure 2) indicate that the error scales as $O(1/n)$ for *any* fixed graph size $|V| = d$. In particular, for a large graph with $d = 100$, `QTree` only needs $n = 200$ observations to bring the metrics nSHD to less than 5% and TPR to more than 95%; see definitions in (1).

We are now ready to state our main theorem. Observe that while α as in (5) is an important tuning parameter, asymptotically it does not matter as the consistency holds for $\alpha = 0$; i.e., by taking the full set of observations.

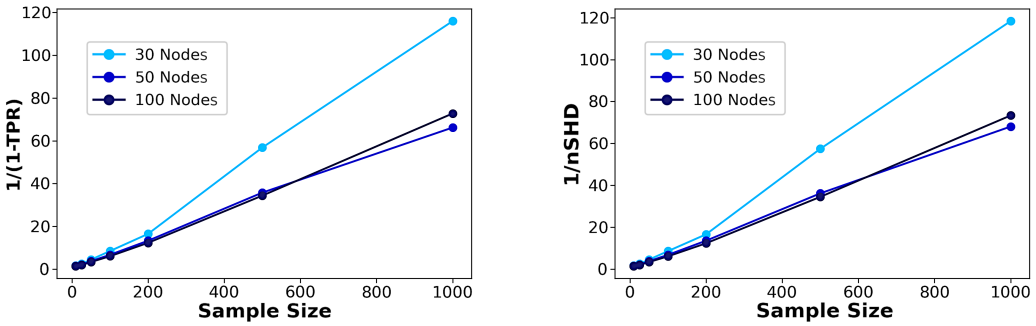

Fig. 2. $1/(\text{mean errors})$ vs. number of observations n for different graph sizes $d = 30, 50, 100$. We simulated 100 root-directed spanning trees as described in Section 5, where we use the Gumbel-Gaussian setting (1). Then we applied `QTree` Algorithm 1 with quantile-to-mean gap (7) with $\underline{r} = 0.05$ and $\alpha = 0$ to estimate the true (simulated) tree, and computed the average error as measured by 1-TPR (left) and nSHD (right) given in (1).

Theorem 1 (Consistency Theorem)**.** *Assume the Gumbel-Gaussian noise model* (8) *with distributions specified above.*
(a) There exists an $r^* > 0$ *such that for any pair* $0 < \underline{r} < \overline{r} < r^*$*, the* `QTree` *algorithm with score matrix* $W = (w_{ij})$ *defined by the lower quantile gap* $w_{ij}(\underline{r}, \overline{r})$ *in* (7) *returns a strongly consistent estimator for the tree* $\mathcal{T}$ *as the sample size* $n \to \infty$*.*
(b) There exists an $r^* > 0$ *such that for any* $0 < \underline{r} < r^*$*, the* `QTree` *algorithm with score matrix* $W = (w_{ij})$ *defined by the quantile-to-mean gap* $w_{ij}(\underline{r})$ *in* (6) *returns a strongly consistent estimator for the tree* $\mathcal{T}$ *as the sample size* $n \to \infty$*.*

Remark 4. [When `QTree` may fail] To understand why a condition like (9) is necessary, suppose that $V = \{1, 2\}$ and that the true graph is $1 \to 2$. Let F_{21} be the distribution function of $(\varepsilon_2 - \varepsilon_1) + (Z_2 - Z_1) \vee c_{21}$. The lower tail of F_{21} essentially is the lower tail of $(\varepsilon_2 - \varepsilon_1)$, while the upper tail is essentially the upper tail of the convolution $(\varepsilon_2 - \varepsilon_1) + (Z_2 - Z_1)$, which is dominated by the signal $(Z_2 - Z_1)$ if it is the heavier tail, and otherwise it is dominated by the noise $(\varepsilon_2 - \varepsilon_1)$. Since $(\varepsilon_2 - \varepsilon_1)$ has symmetric distribution, if the noise term dominates the distribution, $w_{12} \approx w_{21}$ and it would be impossible to distinguish the edge $1 \to 2$ from the edge $2 \to 1$. If the signal dominates, the asymmetry between the lower and upper tails of F_{12} lends us the crucial inequality to distinct between the two graphs as illustrated in Figure 1.

The argument extends to $d > 2$ for a graph with only one directed path. Then it is not possible to distinguish the direction from a symmetric score matrix. However, for a realistic matrix with real-valued entries, Chu–Liu/Edmonds' algorithm outputs an approximately correct root-directed tree. Intuitively, reversing every edge direction gives the same score but is generally not a root-directed tree. □

2.4. Parameter tuning by bootstrap aggegation

Algorithm 1 has two parameters: the quantile $\underline{r} \in (0, 1)$ and the cut-off quantile $\alpha \in (0, 1)$. If data came from a noise-free max-linear Bayesian tree, then we should select $\underline{r} = 0$ as small as possible, $1 - \alpha = 1$, and fit `QTree` on all of the available data $\mathcal{X}$.

Algorithm 2 `Auto-tuned QTree`

Parameters: subsampling fraction $f \in [0,1]$, number of subsamples $m \in \mathbb{N}$, a set of parameters $\Theta = \{(\underline{r}, \alpha)\} \subset [0,1)^2$ to search over.
Input: data $\mathcal{X} = \{x^1, \ldots, x^n\} \subset \mathbb{R}^V$.
Output: the optimal parameter $(\underline{r}^*, \alpha^*) \in \Theta$ and the corresponding root-directed spanning tree $\hat{\mathcal{T}}_{\max}$ on V.

for $(\underline{r}, \alpha) \in \Theta$ **do**
  for $\ell = 1, \ldots, m$ **do**
    Sample without replacement a random subset $\mathcal{X}^\ell$ of $n \cdot f$ observations from $\mathcal{X}$.
    Let $\mathcal{T}^\ell$ be the output of `QTree`$(\underline{r}, \alpha)$ fitted on $\mathcal{X}^\ell$.
  Let $\mathsf{T}(\underline{r}, \alpha) = \{\mathcal{T}^\ell : \ell = 1, \ldots, m\}$
  Compute $S(\mathsf{T}(\underline{r}, \alpha))$ by (12).
  Compute $E(\mathsf{T}(\underline{r}, \alpha))$ as the maximum root-directed spanning tree of $S(\mathsf{T}(\underline{r}, \alpha))$ per Lemma 1.
  Compute $\mathrm{Var}(\mathsf{T}(\underline{r}, \alpha))$ by (11)
Define $(\underline{r}^*, \alpha^*) := \arg\min\{\mathrm{Var}(\mathsf{T}(\underline{r}, \alpha)) : (\underline{r}, \alpha) \in \Theta\}$.
Return the optimal pair $(\underline{r}^*, \alpha^*)$ and $\hat{\mathcal{T}}_{\max} := E(\mathsf{T}(\underline{r}^*, \alpha^*))$.

However, due to the presence of noise, setting $\underline{r}$ too small and $1 - \alpha$ too large would make the estimator volatile to large values of the noise variables.

In this section, we propose in a first step a subsampling procedure to stabilize the tree estimator of Algorithm 1 and, in a second step, automatically choose $\underline{r}$ and α in `QTree`. This results in Algorithm 2, which we also refer to as `auto-tuned QTree`.

The basic idea is to run an algorithm on multiple subsets of the data, and then average the resulting estimator. This subsampling approach is also called bootstrap aggregation or bagging; see (James et al., 2013, Section 8.2.1) and (Politis et al., 1999) for a variety of subsampling procedures. Since `QTree` outputs a directed tree as its estimator, which is a combinatorial object, one cannot simply take the average of their adjacency matrices, as that would not produce a tree. Instead, we see the set of output trees as a distribution over trees. Then, we solve a second problem, namely, to find the centroid tree E(T) of this distribution, defined as that tree which minimizes the expected Hamming distance to a typical tree (cf. Definition 1). Lemma 1 below proves that the centroid can be computed with another application of Chu–Liu/Edmonds' algorithm. This ensures that the estimator produced by auto-tuned `QTree` can be computed quickly (cf. Lemma 2).

Our key indicator for model performance is variability in the estimated tree, that is, whether the tree $\hat{\mathcal{T}}$ and its reachability graph $\hat{\mathcal{R}}$ output by `QTree` would change significantly if we fit it to different subsamples of the data. Here, we denote the reachability graph $\hat{\mathcal{R}}$ of $\hat{\mathcal{T}}$ as the graph that results from drawing an edge between a pair (j, i) whenever there is path from j to i in $\hat{\mathcal{T}}$. We propose the following definition of variability for a distribution of root-directed spanning trees.

Definition 1. Let V be a set of nodes and $\mathsf{T} = \{\mathcal{T}^1, \ldots, \mathcal{T}^m\}$ a collection of root-directed spanning trees on V, and let $\mathsf{R} = \{\mathcal{R}^1, \ldots, \mathcal{R}^m\}$ be their corresponding reachability graphs. The *centroid* of T, denoted $E(\mathsf{T})$, is the root-directed spanning tree on V that

minimizes the sum of normalized structural Hamming distances d_H defined in (1) as follows:

$$E(\mathsf{T}) := \arg\min_{\mathcal{T}\in\Psi} \sum_{i=1}^{m} d_H(\mathcal{T}, \mathcal{T}^i), \tag{10}$$

where Ψ is the space of root-directed spanning trees on V.

Let $E(\mathsf{R})$ denote the reachability graph of $E(\mathsf{T})$. Let e_{T} be the number of edges of $E(\mathsf{T})$, and e_{R} be the number of edges of $E(\mathsf{R})$, respectively. We define the *variability* of T, denoted $\mathrm{Var}(\mathsf{T})$, as

$$\mathrm{Var}(\mathsf{T}) := \frac{1}{e_{\mathsf{T}}}\frac{1}{m}\sum_{i=1}^{m} d_H(\mathcal{T}^i, E(\mathsf{T})) + \frac{1}{e_{\mathsf{R}}}\frac{1}{m}\sum_{i=1}^{m} d_H(\mathcal{R}^i, E(\mathsf{R})). \tag{11}$$

Involving the Hamming distance of the reachability graphs in (11) penalizes the situation where $\mathcal{T}^i$ and $E(\mathsf{T})$ differ in a few edges low down in the tree, for example, if they have different roots. Such a difference would lead to a small structural Hamming distance between the two trees, but a large structural Hamming distance between their reachability graphs, and in particular, very different river networks.

The following lemma says that $E(\mathsf{T})$ is a maximum root-directed spanning tree of a particular graph with score matrix $S(\mathsf{T})$ that measures the stability among the trees in T. In particular, $E(\mathsf{T})$ can be computed using Chu–Liu/Edmonds' algorithm (choosing the root realizing the minimum score), and thus $\mathrm{Var}(\mathsf{T})$ can be computed in polynomial time.

Lemma 1. *Let V be a set of nodes and $\mathsf{T} = \{\mathcal{T}^1, \dots, \mathcal{T}^m\}$ a collection of root-directed spanning trees on V. Define the* stability score matrix *$S := S(\mathsf{T}) \in \mathbb{R}_{\geq 0}^{d\times d}$ by*

$$s_{ij} := S(\mathsf{T})_{ij} := \#\{\mathcal{T} \in \mathsf{T} : j \to i \in \mathcal{T}\}. \tag{12}$$

Suppose that the maximum root-directed spanning tree $\mathcal{T}_{\max}$ of the graph on V with score matrix $S(\mathsf{T})$ is unique. Then $E(\mathsf{T}) = \mathcal{T}_{\max}$.

Proof. Identify a root-directed tree $\mathcal{T}$ with the vector $T = (T_{uv}) \in \{0,1\}^{|V|^2-|V|}$. Write $\mathbf{1} = (\mathbf{1}_{uv})$ for the all-one vector of the same dimension. Let $\mathcal{T}' \in \Psi$ be any root-directed spanning tree on V. Our goal is to show that

$$\sum_{i=1}^{m} d_H(\mathcal{T}', \mathcal{T}^i) \geq \sum_{i=1}^{m} d_H(\mathcal{T}_{\max}, \mathcal{T}^i),$$

which would establish that $\mathcal{T}_{\max} = E(\mathsf{T})$ by (10). Indeed,

$$\begin{aligned}
&\sum_{i=1}^{m} d_H(\mathcal{T}', \mathcal{T}^i) = \sum_{i=1}^{m}\sum_{u,v\in V:u\neq v} \mathbf{1}\{T'_{uv} \neq T^i_{uv}\} = \sum_{u,v\in V:u\neq v}\sum_{i=1}^{m} \mathbf{1}\{T'_{uv} \neq T^i_{uv}\} \\
&= \sum_{u,v\in V:u\neq v} (s_{uv}\mathbf{1}_{u\to v\notin\mathcal{T}'} + (m - s_{uv})\mathbf{1}_{u\to v\in\mathcal{T}'}) \\
&= -2\langle S, T'\rangle + \langle S, \mathbf{1}\rangle + \langle m\mathbf{1}, T'\rangle \quad \text{where } \langle\cdot,\cdot\rangle \text{ denotes the Frobenius inner product}
\end{aligned}$$

$$
\begin{aligned}
&= -2\langle S, T' \rangle + \langle S, \mathbf{1} \rangle + m(d-1) \quad \text{since } \mathcal{T}' \text{ as root-directed spanning tree has } d-1 \text{ edges} \\
&\geq -2\langle S, T_{\max} \rangle + \langle S, \mathbf{1} \rangle + m(d-1)\rangle \quad \text{by definition of } \mathcal{T}_{\max} \\
&= -2\langle S, T_{\max} \rangle + \langle S, \mathbf{1} \rangle + \langle m\mathbf{1}, T_{\max} \rangle\rangle \quad \text{since } \mathcal{T}_{\max} \text{ is a spanning tree on } V \\
&= \sum_{i=1}^{m} d_H(\mathcal{T}_{\max}, \mathcal{T}^i).
\end{aligned}
$$

□

As shown in Lemma S2 of the Supplementary Material, Algorithm 1 runs in time $O(|V|^2 n)$. The quadratic dependence on $|V|$ and linear dependence on n is optimal, since it takes $O(|V|^2 n)$ just to compute pairwise statistics such as the concentration measures in (6) or (7) for every pair of nodes. Similarly, the runtime of Algorithm 2 (auto-tuned `QTree`) also has optimal runtime, which scales linearly with the number of repetitions m and the size of the parameter grid $|\Theta|$.

Lemma 2. *The* ***auto-tuned*** `QTree` *Algorithm 2 has complexity* $O(|V|^2 nm|\Theta|)$.

Proof. For each pair $(\underline{r}, \alpha) \in \Theta$, step 3 takes $O(mn)$ and step 4 takes $O(|V|^2 nm)$ by Lemma S2 in the Supplementary Material. Step 6 takes $O(m|V|^2)$, and step 7 takes $O(|V|^2)$ by Chu–Liu/Edmonds' Algorithm. Computing the reachability graph for a root-directed tree on $|V|$ nodes takes $O(|V|)$, so step 8 takes $O(m|V|^2)$, since for each of the m trees in T we need to compute its structural Hamming distance from the estimated tree $E(\mathsf{T})$. So, for each pair $(\underline{r}, \alpha) \in \Theta$, steps 3 to 8 take $O(|V|^2 nm)$ time. Thus overall, the algorithmic complexity is $O(|V|^2 nm|\Theta|)$. □

3. Data description

We focus on river discharge data in two river networks, the Upper Danube network with data from Bavaria, Germany, and the Lower Colorado network in Texas, USA. Large flood events are classical examples for high risk analysis. The Danube data as well as the data of all three sectors of the Colorado are available in the `Python` package `QTree` (Tran (2021)). The Danube data are available in the `R` package `graphicalExtremes` (Engelke et al. (2019)).

In general, river discharges across a set of stations is recorded multiple times per hour and some preprocessing is needed to turn the raw data into independent extreme discharge data. This was detailed in Asadi et al. (2015) for the Danube data; cf. Figure 3. We follow their procedure (descibed in Section 3.1) with slight modifications for the Colorado data (descibed Section 3.2).

3.1. The Upper Danube network

The Danube network data consist of measurements collected at $d = 31$ gauging stations over 50 years from 1960 to 2009 by the Bavarian Environmental Agency (`http:www.gkd.bayern.de`). Preprocessing the data, Asadi et al. (2015) first take daily mean values in each time series. The idea is then to find non-overlapping time windows of p days, centered around the observation of maximal rank across all series. For the Danube, the

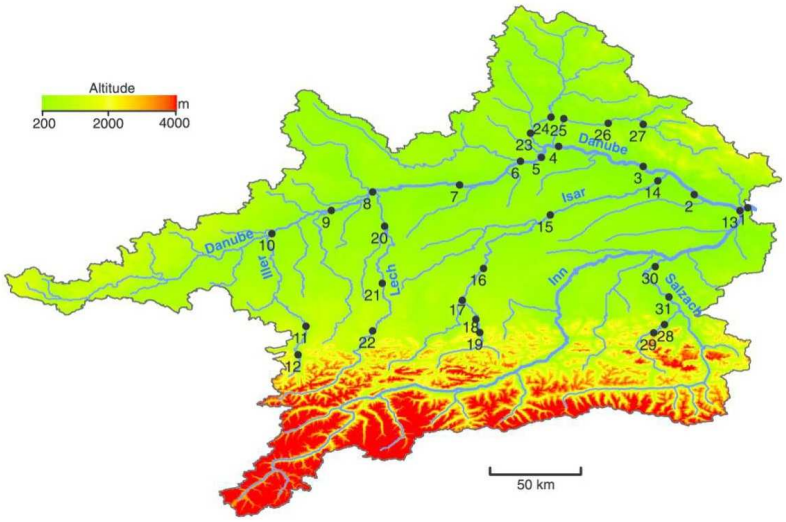

Fig. 3. Topographic map of the Upper Danube Basin, showing sites of the 31 gauging stations along the Danube and its tributaries.

authors choose $p = 9$ days (± 4 days around the observation of maximal rank). For each time series, they then take the maximum within the given time window, delete the data of this window, and proceed until no window of p consecutive days remains. In order to reduce temporal non-stationarities and the effect of snow melt, only the months June, July and August are considered. This results in $n = 428$ observations from a $d = 31$-dimensional random vector whose i-th entry corresponds to the maximum water discharge at the i-th station, observed within a 9-day window where at least one station witnessed a large discharge value; these observations are assumed to be independent.

3.2. The Lower Colorado network in Texas

This section describes the new data set of the Lower Colorado river network in Texas collected by the Lower Colorado River Authority (LCRA, `https://www.lcra.org/`) and details the preprocessing.

The Lower Colorado is one of the major rivers in Texas. Flowing through major population centers such as Austin, the state capital of Texas, flood and drought mitigation in the Lower Colorado Basin is of prominent interest. A particularly challenging feature of the Lower Colorado is prolonged drought (discharge of 0) followed by flash flooding which can damage sensors, resulting in loss of data over multiple days (cf. Figure 4). This makes the Lower Colorado data much more challenging than the Danube data.

The river discharges at the Lower Colorado network, measured in cubic feet per second (cfs), are collected multiple times per day at a total of 104 stations around the Colorado River and its tributaries in Texas from the 1st of December 1991 to the 14th of April 2020 (10,363 days); see Figure 5. We do not take into account 5 nodes of the Blanco River and San Bernand River, which are not flow-connected to the Lower Colorado River, and also 21 nodes with zero observations. Moreover, we exclude the nodes 5476, 5634, 5635, 6397, and 6533 as they are located close to hydropower plants. This gives a total of 73 nodes.

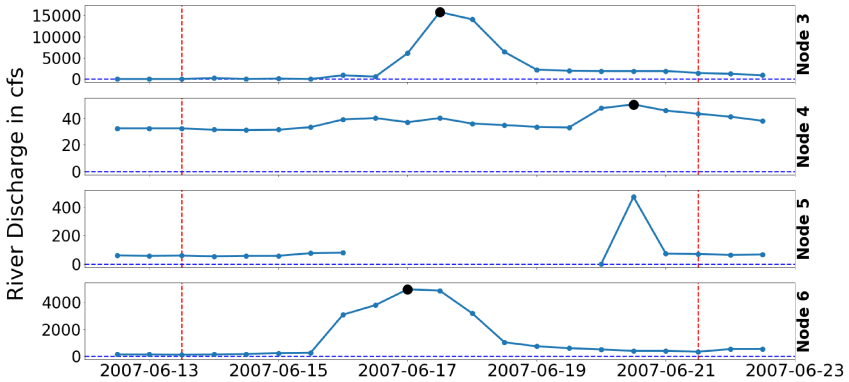

Fig. 4. Typical discharge at various nodes around one flood event on the Lower Colorado. The vertical lines mark a time window of $p = 17 \times 12h$ or $p = 8.5$ days. Dots denote the 12-hour maximum water discharges. In nodes 3,4 and 6, the thick dot denotes the peak discharge during this window. For node 5, the sensor did not function during this entire time period, so the peak for node 5 is recorded as missing. Node 3 has a median discharge of only 15 cfs and is hence mostly drained over the entire period (1991 to 2020, 10,363 days), but the water flow regularly aggregates to over 16,000 cfs within a very short period of time.

Another problem occurs, because in the Lower Colorado Basin, multiple dams cut off the river into disjoint sectors (Lower Colorado River Authority (LCRA), 2020a). Thus, we split the river network such that in each section, we get the largest set of nodes where (i) no node is within 10km of a major dam, (ii) all nodes are connected, and (iii) for each pair (j, i) of this subset, there are at least 1000 pairwise daily observations, which is 9.6% of the total amount of 10,363 days available. Criterion (iii) ensures that among the given pairs of nodes, any possible causal relation can be discovered, and not be affected by the lack of concurrent data. This results in 42 nodes divided into three sectors, which we call the *Top, Middle and Bottom* sectors of the Lower Colorado with 9, 12 and 21 nodes, respectively. From here on we treat these three sectors as three separated, unrelated data sets.

In contrast to the Danube network with snow melt and seasonal periodicity, we can and do take all data of the 12 months into account. We observe further that, by the special weather conditions, flood events can last as little as a few hours. Therefore, in contrast to Asadi et al. (2015), who take daily time slots, we take 12-hour time slots. As a first step, we take maxima of each 12-hour time slot to retain the knowledge about large possible peaks and such periods where no data are collected. In a second step we then take ± 8 12-hour time slot around the time-slots with the observation of maximal rank resulting in a time window of p=8.5 days; see Figure 4. We take the most conservative approach to missing data, namely, if node i has any missing data during the considered time window, then its maximum discharge over this window is labeled as missing (cf. Figure 4). This is because a sensor can break before the river reaches peak discharge and for practical reasons can only be replaced after the flood event is over (Lower Colorado River Authority (LCRA), 2020b), and thus the sensor potentially did not measure the largest possible water discharge that occurred at node i. This results in the Top sector having 9 nodes, 975 observations, 18% missing data; the Middle sector has 12 nodes, 972 observations, 27% missing data. The Bottom sector is most challenging, for it has the

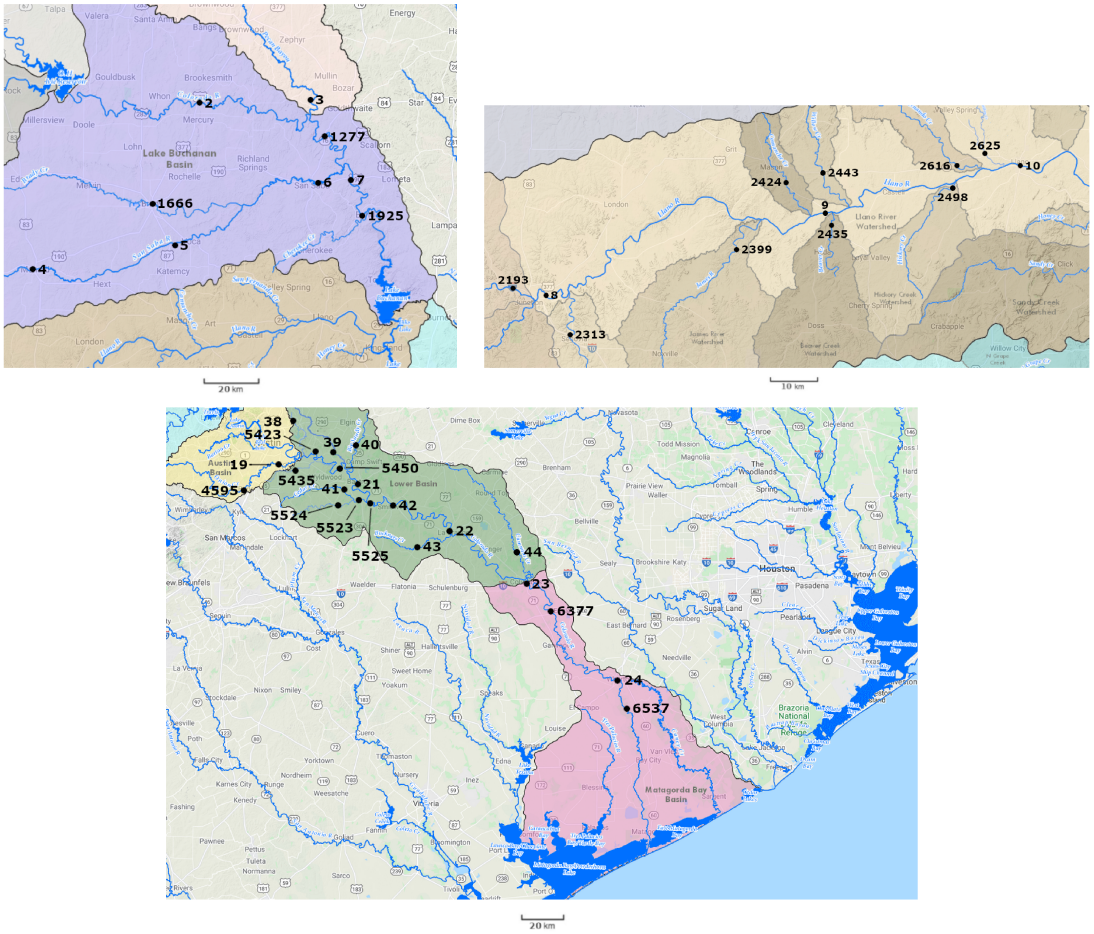

Fig. 5. Topographic maps of the Top, Middle and Bottom sectors (arranged clockwise) of the Lower Colorado network, showing sites of the gauging stations along the Colorado River and its tributaries. We treat them as three unrelated data sets.

most nodes (21 nodes), 961 observations, the highest amount of missing data (37 %), and many nodes around the city of Austin with only a few miles apart from each other. In the Bottom sector there are many nodes with a very small number of observations due to the many missing observations, and we create a new data set by excluding all nodes with less than 150 observations and refer to them as *Bottom150*.

The close proximity of nodes induces strong spatial dependence even among nodes that are not flow-connected, making it potentially more challenging to recover the true network. A summary of the abailable data for each data set is given in Table 1.

4. Results

4.1. *Results of* `auto-tuned QTree` *for all river networks*

For each of the four river networks (Danube, Top, Middle and Bottom sectors of the Colorado), we ran `auto-tuned QTree` (Algorithm 2) with fixed $\underline{r} = 0.05$, subsampling

Table 1. Number of nodes d, number of observations n and percentage of missing data used for the algorithmic reconstruction of the river network.

	Danube	Top	Middle	Bottom	Bottom150
d	31	9	12	21	16
n	428	975	972	961	961
%	0%	18%	27%	37%	22%

Table 2. Optimal parameters α^* selected by `QTree` using grid search with subsampling.

	Danube	Top	Middle	Bottom	Bottom150
α^*	0.775	0.825	0.75	0.85	0.725

rate $f = 0.75$, and number of repetitions $m = 1000$ to choose the tuning parameter α automatically from $\{0.7, 0.725, 0.75, , \ldots, 0.9\}$. The optimal parameters α^* selected by `QTree` for these networks are shown in Table 2.

Figures 6—8 show the estimated trees of the Danube, Top, Middle and Bottom sectors of the Colorado, respectively. We do two estimated-vs-true comparisons: one for the tree, and one for its reachability graph. The four metrics we use are normalized Structural Hamming Distance (nSHD), False Discovery Rate (FDR) and False Positive Rate (FPR) and True Positive Rate (TPR), defined in (1). Table 3 gives all performance metrics over all data sets. We recall that the performance of an algorithm is better the smaller the first three metrics are and the larger TPR is. `QTree` performs very well across all data sets, with nSHD and FDR ranging from 10-20%, FPR close to 0, and TPR around 80-90%. For the reachability graph, the statistics are even better: nSHD, FPR and FDR are below 9% and TPR is over 87%. In other words, a wrongly estimated edge directs rather from an ancestor (which is not a parent) to a child (flow-connection is preserved), than a spurious edge (an edge which contradicts flow-connection). The number of missing edges are determined by the fact that a tree has exactly $d - 1$ edges.

Figure 8(top) visualizes the estimation of the Bottom sector of the Colorado. As expected, this data set is the most challenging due to large portions of missing data and the clustering of nodes around the city of Austin. Nevertheless, even for this data set the estimated tree has only two spurious edges, between 6537 and 24 and between 42 and 5525. Both of these node pairs are physically close. All the remaining wrongly estimated edges are flow-connected.

We note that the majority of errors made by `QTree` involves nodes with less than 150 observations (which are the nodes 5525, 5450, 5423, 5435 and 5524). This is not at all surprising. The model was fitted to only 75% of the data, and the optimally chosen α^* is 0.85, which means that for each edge involving one of the above nodes, the number of observations available to `QTree` is at most $150 \times 0.75 \times 0.15 = 14$. To check the hypothesis that this threshold is too small for `QTree` to perform reliably, we excluded all nodes with less than 150 observations and refitted `QTree` on the remaining 16 nodes (Bottom150). The result depicted in Figure 8(bottom) shows significant improvements. This manifests another desirable feature of `QTree`, namely, that it relies on local (pairwise) estimation, and thus changes to the node set in one part of the tree do not affect the estimated network elsewhere.

Table 3. Metrics nSHD, FPR, FDR and TPR for the `QTree`. Numbers display the respective metric for the pair $(\mathcal{T}, \hat{\mathcal{T}})$ and numbers in brackets for the pair $(\mathcal{R}, \hat{\mathcal{R}})$ of their respective reachability graphs.

	Danube	Colorado			
		Top	Middle	Bottom	Bottom150
nSHD	0.18(0.09)	0.13(0.02)	0.09(0.06)	0.45(0.15)	0.10(0.12)
FPR	0.01(0.02)	0.04(0.00)	0.02(0.04)	0.05(0.03)	0.02(0.02)
FDR	0.20(0.05)	0.13(0.00)	0.09(0.10)	0.50(0.02)	0.13(0.02)
TPR	0.80(0.87)	0.88(0.95)	0.90(1.00)	0.50(0.74)	0.87(0.78)

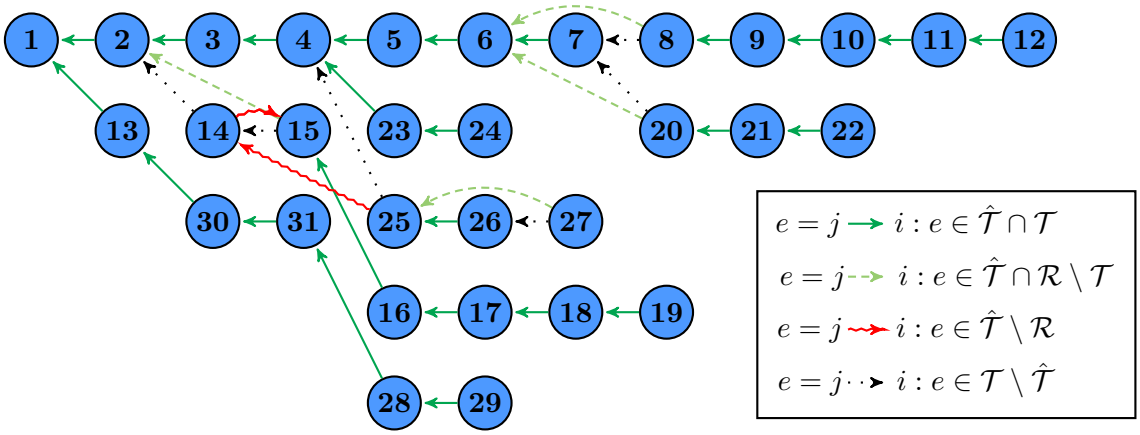

Fig. 6. Danube river network, estimated by `QTree` vs. true. Solid (green) edges are correct. Dashed (green) edges are not in the tree but in the reachability graph, that is, the causal direction or flow-connection is correct. Squiggly (red) edges are spurious (neither in the tree nor in the reachability graph). Dotted (black) edges are in the true tree, but not in the estimated tree. `QTree` outputs a tree with only six wrongly estimated edges, four of them flow-connected and one path skipping a single node (the edge $8 \to 6$ skips node 7). Two edges are spurious.

As expected from a statistical estimation procedure, the statistical choice of the parameter selection by `QTree` does not always output the best result on every data set. However, it fails only by very few edges to the graph estimated with the best choice of parameters. For example, for the Danube the parameter $\alpha = 0.75$ (instead of the optimal $\alpha^* = 0.775$) would have lead to a better result (cf. Figure 9). Also for the Top Colorado, $\alpha = 0.9$ would have given perfect recovery of the true network; this is clear as all four metrics become optimal (Figure S4 of the Supplementary Material).

In summary, on all four data sets considered, `QTree` performed well for nodes with a sufficient number of observations as becomes obvious from Figure 8(top) and (bottom). The estimated optimal parameter α^* is either the best one (i.e., the corresponding estimated tree is best across all α) (Top and Bottom sectors of the Colorado), or such that it is within one to two wrong edges of the best one (Danube and Top sector of the Colorado). The method can handle data with missing observations and close spatial proximity between nodes.

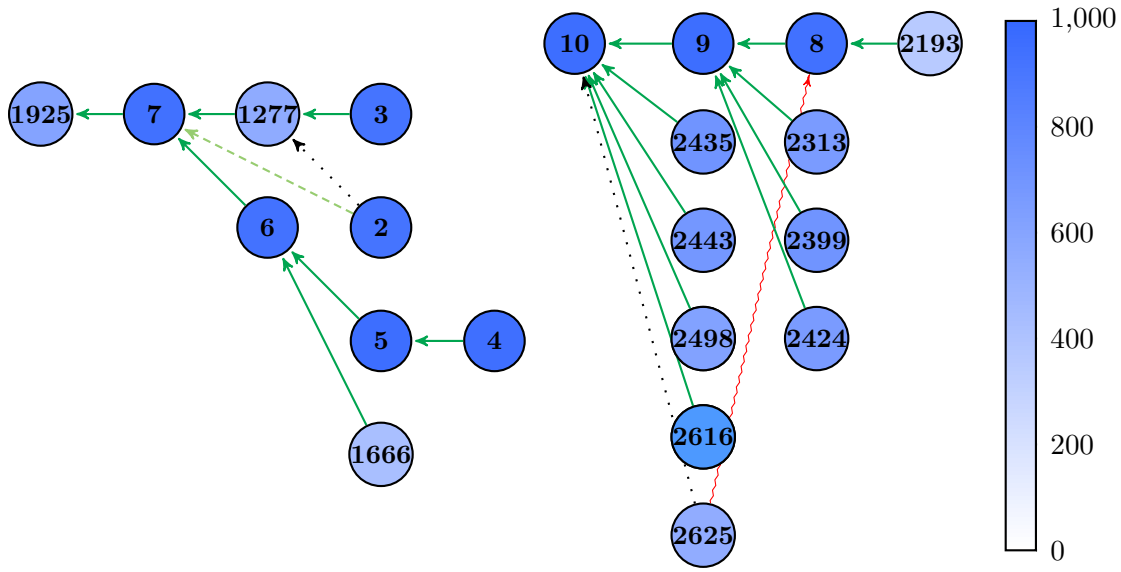

Fig. 7. Top (left) and Middle (right) sectors of the Colorado network, estimated by `QTree` vs. true. Node colors represent the amount of available data after taking care of missing data. Arrows are as described in Figure 6. Both estimated networks only contain one single wrongly estimated edge. Top: one edge wrong but flow-connected; Middle: one edge spurious.

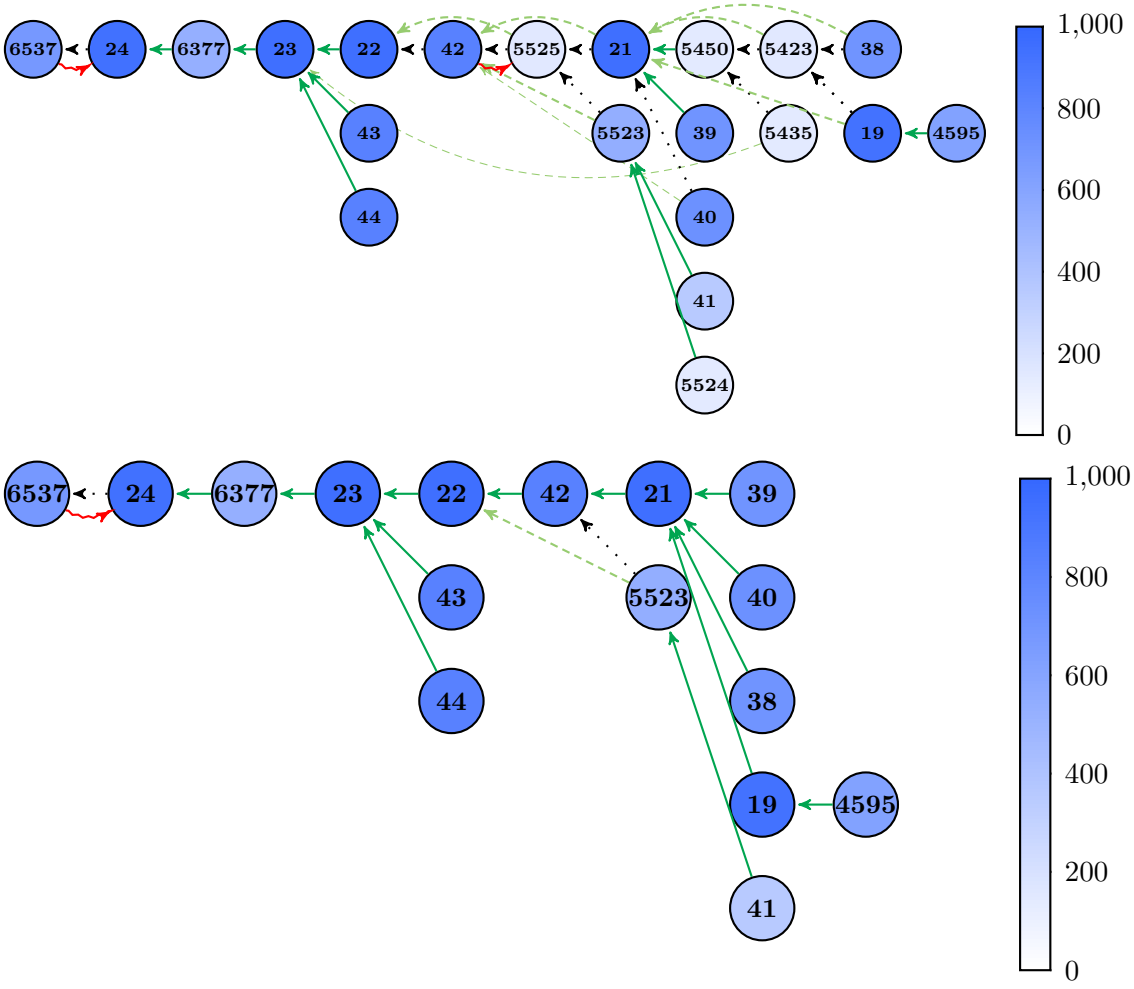

Fig. 8. Bottom sector of the Colorado network (Bottom and Bottom150), estimated by `QTree` vs. true. Top Figure: Bottom, based on all 21 nodes, `QTree` outputs a tree with ten wrongly estimated edges, eight of them flow-connected, two spurious edges pointing in the wrong direction. Bottom Figure: Bottom150, based on 16 nodes, after removing nodes with less than 150 observations. There are only two wrongly estimated edges, one flow-connected, one spurious edge pointing in the wrong direction. Compared to the Bottom sector, this is a significant improvement. Node colors represent the amount of available data after taking care of missing data. Arrows are as described in Figure 6

.

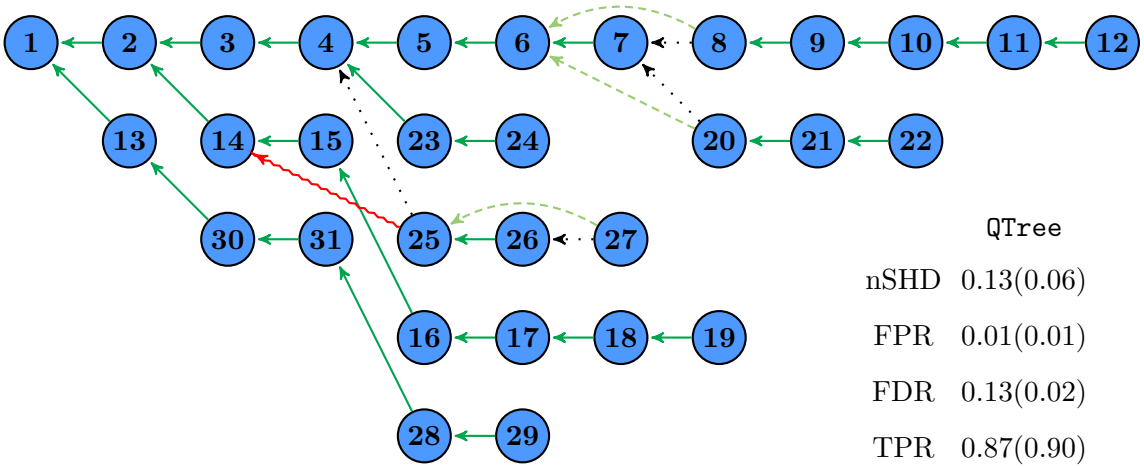

Fig. 9. Danube river network, estimated by `QTree` vs. true for $\alpha = 0.75$. Compared to Figure 6, the edges $15 \to 14$ and $14 \to 2$ are here correctly estimated. Also the performance measures at the right bottom of the figure compare favourably to those in the first row of Table 3.

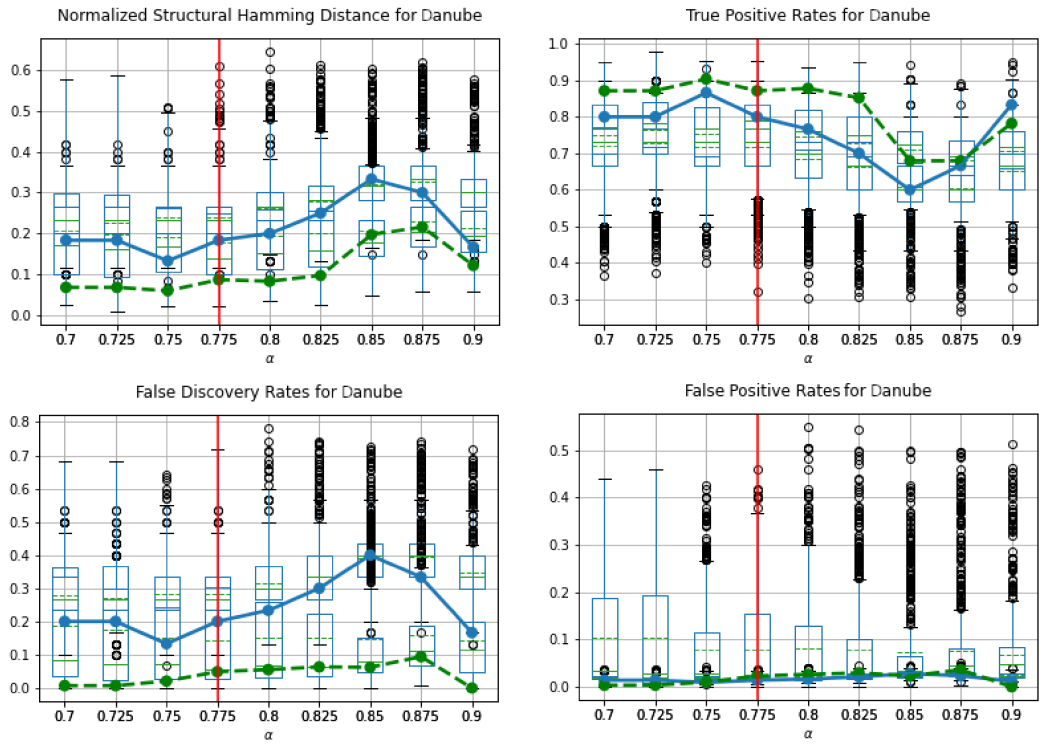

Fig. 10. Metrics nSHD, TPR, FDR and FPR for the output $\hat{\mathcal{T}}_\alpha$ of the steps of `auto-tuned QTree` for varying parameter α for the Danube network. We subsample 75% of the data 1000 times. For each α we fit `QTree` on these subsamples to obtain 1000 estimated trees $\mathsf{T}_\alpha := \{\hat{\mathcal{T}}_\alpha^1, \ldots, \hat{\mathcal{T}}_\alpha^{1000}\}$ (Step 5 of Algorithm 2). The metrics of $(\hat{\mathcal{T}}_\alpha^\ell, \mathcal{T})$ and $(\hat{\mathcal{R}}_\alpha^\ell, \mathcal{R})$ are represented in boxplots, one for the tree (blue) and one for the reachability graph (green). The blue and green dots present the four metrics for the centroid $E(\mathsf{T}_\alpha)$ (Step 7 of Algorithm 2) and its reachability graph. The lines are interpolations for better visibility, solid blue for the tree and dashed green for the reachability graph. The chosen α^* is the parameter with the least variablity in T_α (Step 9 of Algorithm 2), indicated by a red vertical line.

4.2. Comparison to other scores in the literature

We compare `QTree` and `auto-tuned QTree` with existing algorithms for extremal causal estimation in the literature. To this end, we first define the scores and summarize them in score matrices. We utilize the empirical versions of the following extreme dependence measures, where we also include the quantile-to-mean gap for comparison:

(a) The empirical *quantile-to-mean gap* as in (6) for fixed $(\underline{r}, \alpha) = (0.05, 0.9)$. We fix $\underline{r} = 0.05$ as we have used this throughout, and $\alpha = 0.9$ as an arbitrary parameter.

(b) The *causal tail coefficient* $\Gamma_{ij} = \lim_{u\to 1} \mathbb{E}[F_i(X_i) \mid F_j(X_j) > u]$ (Gnecco et al., 2021, eq. (3)). Observe that in Gnecco et al. (2021), the algorithm EASE outputs from the estimated score matrix Γ an estimated causal order of the set of nodes, not a directed graph.

(c) The *causal score* S_{ij}^{ext} is based on expected quantile scores (Mhalla et al., 2020, eq. (15)).§ The goal of the authors is to discover causality in a directed graph modelled by flow-connection. The scores S_{ij}^{ext} satisfy $S_{ji}^{\text{ext}} + S_{ij}^{\text{ext}} = 1$ and an extreme observation at node j causes an extreme observation at node i, whenever $S_{ij}^{\text{ext}} > 0.5$. The authors propose a bootstrap method to generate 95% confidence bounds to guarantee that the score is larger than 0.5. All these scores are interpreted as directed edges between nodes. Also for the tree example of the Danube data treated in Section 5 of the paper (cf. Figure 7). The algorithm CausEV outputs flow-connections induced by all scores larger than 0.5. Thus, their causal edges rather resemble the edges in the reachability graph of the tree.

(d) The *tail dependence coefficient* $\chi_{ij} = \lim_{u\to 1} P(F_i(X_i) > u \mid F_j(X_j) > u)$, also called *extremal correlation*. It goes back to Sibuya (1960); see also (Coles et al., 1999). We add this dependence measure in our comparison as it is *the* classic one, having been used for more than 60 years in multivariate extreme value statistics. Theoretical properties of the tail dependence coefficient in a max-linear Bayesian network have been investigated in Gissibl et al. (2018). It has values in $[0, 1]$ and a large value of χ_{ij} indicate strong extreme dependence. Its empirical estimator takes u large, but finite, and our estimator is based on 10% of the data (corresponding to $\alpha = 0.9$ in (a)). The paper Engelke and Volgushev (2020) uses χ and two related measures on p. 14, the extremal correlation, the extremal variogram and the combined extremal variogram for estimating undirected trees with Prim's algorithm Prim (1957).

The scores discussed (b)-(d) have not been used to estimate a directed tree, but as they are all pair-wise scores, their respective estimated matrices can serve as input for Chu–Liu/Edmonds' algorithm. This gives a fair comparison, as we use then for all scores the knowledge that the true graph is a root-directed spanning tree. A few words on the estimation of the matrices Γ and S^{ext} are necessary to fully understand the estimation procedure. For better comparability, we estimate the score matrix Γ also for the declustered Danube data. For the causal score S_{ij}^{ext} we recall that a spanning tree with d nodes has exactly $d-1$ edges, which are taken by Chu–Liu/Edmonds' algorithm as large as possible under the restriction that the outcome is a spanning tree. The

§We want to thank Linda Mhalla for helping us to set up the CausEv implementation.

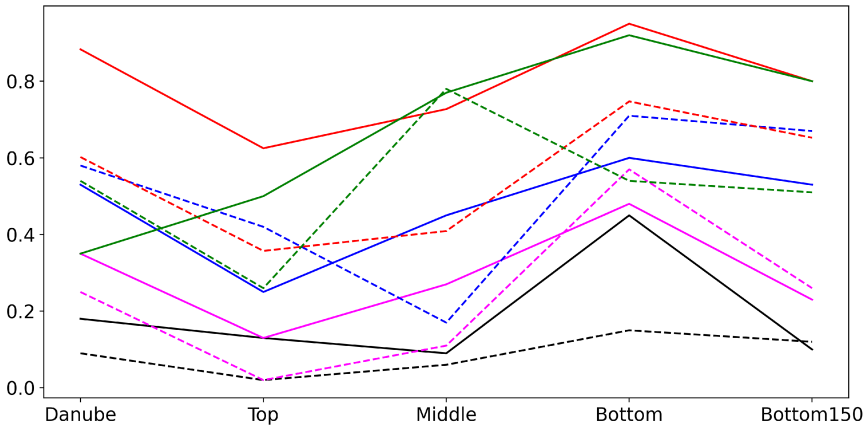

Fig. 11. Performance metric nSHD for all five data sets and scores vs. `QTree`: `auto-tuned QTree`, `QTree`, S^{ext} as in Mhalla et al. (2020), Γ as in Gnecco et al. (2021) and χ as in Sibuya (1960). Solid lines display the respective metrics for the pair $(\mathcal{T}, \hat{\mathcal{T}})$ and dashed lines for the pair $(\mathcal{R}, \hat{\mathcal{R}})$ of their respective reachability graphs. Black lines are used for `auto-tuned QTree`, magenta lines for `QTree`, red lines for S^{ext}, blue lines for Γ and green lines for χ.

estimated root-directed spanning tree has smallest score of 0.5183, thus, it is above 0.5, so all directions are causal in the sense of Mhalla et al. (2020). While the matrix χ is symmetric, causal inference is possible with Chu–Liu/Edmonds' algorithm. This is because reversing every edge direction gives the same score but is generally not a root-directed spanning tree; see Remark 4. Finally, we keep all tuning parameters used for Γ_{ij} and S^{ext}_{ij} the same as in the respective papers.

Hence, our comparison is two-fold.

(a) We compare the new score of a quantile-to-mean gap with other scores from the literature.
(b) We compare `QTree` with `auto-tuned QTree` assessing the benefit of the stabilizing subsampling procedure.

Tables 5, 6 and 7 present the metrics nSHD, FPR, FDR and TPR based on the three scores (b)—(d) and for each data set previously considered. Comparing the metrics with those of `QTree` in Table 4, we find for the Danube network a comparably weak performance for all three alternative scores. Among the three scores, χ performs best, followed by Γ. We visualize our findings from Table 3 and Tables 4—7 for nSHD in Figure 11.

Solid lines present nSHD for the estimated tree vs. true tree. We find that `auto-tuned QTree` (black line) is uniformly best over all data sets, followed by `QTree`. We conclude that the stabilizing subsampling procedure of `auto-tuned QTree` improves the estimated tree. Even for the Top Colorado network, where $\alpha = 0.9$ is optimal, the nSHD is still positive for `QTree` whereas for `auto-tuned QTree`, nSHD is equal to zero and the estimated tree is equal to the true one. For all Colorado sectors, Γ follows next, χ is surprisingly successful for the Danube, but not for any of the Colorado sectors. S^{ext} does not perform well in any tree recovery, but this was also not the goal of Mhalla et al. (2020)

Table 4. Metrics nSHD, FPR, FDR and TPR for the simple `QTree` Algorithm 1 with $\alpha = 0.9$. Numbers display the metrics for the pair $(\mathcal{T}, \hat{\mathcal{T}})$ and numbers in brackets for the pair $(\mathcal{R}, \hat{\mathcal{R}})$ of their respective reachability graphs.

	Danube	Colorado			
		Top	Middle	Bottom	Bottom150
nSHD	0.35(0.25)	0.13(0.02)	0.27(0.11)	0.48(0.57)	0.23(0.26)
FPR	0.01(0.02)	0.02(0.00)	0.02(0.03)	0.03(0.15)	0.02(0.01)
FDR	0.40(0.14)	0.13(0.00)	0.27(0.19)	0.60(0.58)	0.27(0.02)
TPR	0.60(0.64)	0.88(0.95)	0.72(1.00)	0.40(0.23)	0.73(0.59)

Table 5. Metrics nSHD, FPR, FDR and TPR presented as in Table 3 for the maximum root-directed spanning tree estimated by Chu–Liu/Edmonds' algorithm with score matrix Γ as in Gnecco et al. (2021), eq. (8).

	Danube	Colorado			
		Top	Middle	Bottom	Bottom150
nSHD	0.53(0.58)	0.25(0.42)	0.45(0.17)	0.60(0.71)	0.53(0.67)
FPR	0.02(0.18)	0.05(0.12)	0.04(0.03)	0.04(0.17)	0.04(0.27)
FDR	0.73(0.71)	0.38(0.40)	0.45(0.21)	0.70(0.78)	0.67(0.83)
TPR	0.27(0.37)	0.63(0.43)	0.55(0.88)	0.30(0.10)	0.33(0.12)

Table 6. Metrics nSHD, FPR, FDR and TPR presented as in Table 3 for the maximum root-directed spanning tree estimated by Chu–Liu/Edmonds' algorithm with score matrix S^{ext} as in Mhalla et al. (2020), eq. (15).

	Danube	Colorado			
		Top	Middle	Bottom	Bottom150
nSHD	0.88(0.60)	0.63(0.36)	0.73(0.41)	0.95(0.75)	0.80(0.65)
FPR	0.03(0.10)	0.08(0.18)	0.07(0.12)	0.05(0.12)	0.05(0.17)
FDR	0.90(0.60)	0.63(0.43)	0.73(0.52)	0.95(0.68)	0.80(0.68)
TPR	0.10(0.33)	0.38(0.57)	0.27(0.76)	0.05(0.12)	0.20(0.17)

Table 7. Metrics nSHD, FPR, FDR and TPR presented as in Table 3 for the maximum root-directed spanning tree estimated by Chu–Liu/Edmonds' algorithm with score matrix χ as in Sibuya (1960) or Coles et al. (1999).

	Danube	Colorado			
		Top	Middle	Bottom	Bottom150
nSHD	0.35(0.54)	0.50(0.26)	0.77(0.78)	0.92(0.54)	0.80(0.51)
FPR	0.02(0.21)	0.06(0.25)	0.07(0.33)	0.05(0.27)	0.06(0.30)
FDR	0.47(0.69)	0.50(0.45)	0.82(0.93)	0.95(0.69)	0.87(0.69)
TPR	0.53(0.48)	0.50(0.76)	0.18(0.18)	0.05(0.26)	0.13(0.28)

Dashed lines present nSHD for the reachability matrix of the estimated trees vs. true. Here `auto-tuned QTree` and `QTree` give the same answers as for the trees above. The blue dashes line representing Γ is only moderately worse than the magenta line for `QTree` for the Middle and Bottom sector of the Colorado. χ is worst for the Middle Colorado, for the Danube and the other sectors of the Colorado it performs better than Γ and S^{ext}.

We conclude that for the Danube as well as for the various sectors of the Colorado, `auto-tuned QTree` outperforms uniformly all algorithms without the stabilizing subsampling procedure; see Figure 11. Moreover, the `QTree` score outperforms the other scores when applying Chu-Liu/Edmonds' algorithm, therefore we conclude that the quantile-to-mean gap (6) is superior to the other scores on all data sets considered.

5. A small simulation study

Our main result, Theorem 1 ensures strong consistency of the output trees of `QTree` when the sample size n tends to infinity. In this section we show the quality of `QTree` through the two metrics nSHD and TPR for varying n and d by a small simulation study.

We generate data $\mathcal{X}$ from a max-linear Bayesian tree as defined in equation (3) with $|V| = d$ nodes. For each node i, we calculate the sample standard deviation $\hat{\sigma}_{X_i}$ of $(X_i^1, \dots, X_i^n)$ and take the sample median $\hat{\sigma}$ over all nodes $1, \dots, d$. We then generate i.i.d. normally distributed noise variables ε_i^t with mean zero and standard deviation $k \cdot \hat{\sigma}$ for $i \in V$ and $t = 1, \dots, n$. For the *noise-to-signal ratio* k, we choose $k = 30\%$.

We generate root-directed spanning trees as follows. We first generate a random undirected spanning of size d using the graph generators module `networkX` (Release 2.8.8) in `Python` (Hagberg et al., 2008). We then choose the root node uniformly at random which uniquely determines the root-directed spanning tree. Finally, we assign edge weights c_{ij} independently.

For the distributions of the innovations $Z_1, \dots, Z_d$ and the edge weights c_{ij}, we consider the following three settings:

(1) Innovations $Z_1, \dots, Z_d$ are independent Gumbel$(1, 0)$ distributed and for every edge, we draw an edge weight c_{ij} from the interval $[\log(0.1), \log(1)]$ uniformly. We refer to this as the *standard Gumbel setting*.
(2) Innovations $Z_1, \dots, Z_d$ are independent Gumbel$(1, 0)$ distributed and for every edge, we draw an edge weight c_{ij} from the interval $[\log(0.1), \log(0.3)]$ uniformly. We refer to this as the *weak dependence setting*.
(3) 50% of the innovations $Z_1, \dots, Z_d$ are Gumbel$(1, 0)$ and 50% are $\mathcal{N}(0, 1)$. For every edge, we also draw an edge weight c_{ij} from the interval $[\log(0.1), \log(1)]$ uniformly. We refer to this as the *mixed distribution setting*.

For the score, we take the quantile-to-mean gap as in (6) (normalization by n_{ij} is not needed in a simulation setting) given by

$$w_{ij}(\underline{r}) := \left(\mathbb{E}(\mathcal{X}_{ij}(\alpha)) - Q_{\chi_{ij(\alpha)}}(\underline{r})\right)^2$$

and apply the `QTree` Algorithm 1 with parameters $\underline{r} = 0.05$ and $\alpha = 0$; we remark that a sensitivity analysis has shown that altering $\underline{r}$ influences the results only insignificantly.

We use graph sizes $d = 10, 30, 50, 100$ and 100 repetitions. For each repetition, we calculate the normalized Structural Hamming Distance (nSHD) and the True Positive Rate (TPR), and then take the mean over all 100 repetitions. For definitions of these metrics, see equation (1). Observe that both metrics are normalized to lie in the interval $[0, 1]$ and for nSHD smaller values are better, whereas for TPR larger values are better.

As `QTree` performs so well not only on simple data like those from the Danube network, but also on all sectors of the Colorado network, we guess that it is fairly robust towards the strength of dependence, given by the c_{ij} and even different node distributions. The weak dependence setting (2) should manifest whether `QTree` is also able to recover the underlying network if the dependence given by the weights c_{ij} is much much smaller. We want to quantify robustness towards node distributions with the mixed distribution setting (3).

Figures 12 and 13 depict the mean nSHD and TPR standard Gumbel setting (1) and all four graph sizes. Both metrics quickly tend to zero, respectively one, as the sample sizes n increase. Moreover, comparing the four subfigures for a fixed sample size n, the metrics perform only slightly worse for increasing graph size d.

Figures S8 and S9 in Section S4 of the Supplementary Material depict the mean nSHD and TPR for the weak dependence setting (2) and all four graph sizes. Again, both metrics quickly tends to zero, respectively one. In comparison to the previous setting (1), it performs slightly worse.

Figures S10 and S11 in Section S4 of the Supplementary Material depict the mean nSHD and TPR for the mixed distribution setting (3) and all four graph sizes. Despite the different distributions of the innovations, both metrics quickly tend to zero, respectively one. The performance compared to settings (1) and (2) is expectedly worse, however, less than perhaps could be expected.

The decrease in performance for increasing graph size d is presented in Figure 14, where we plot the minimum amount of data n needed to reach a mean nSHD of 10%. The first observation is that larger networks need a larger sample size to reach a nSHD below 10%. Since larger networks have more opportunities for a wrongly estimated causal influence, this is in line with what we expect. Moreover, the standard Gumbel setting (1) converges much faster than both other settings. Although the weights c_{ij} of setting (2) are in general much smaller than the weights of setting (1), only for a very large graph, substantially more data are needed to reach the lower bound for the nSHD. This implies that the smaller dependence impacts the estimation for a small graph only moderately, but for a larger graph more data are required. The mixed distribution setting (3), however, requires for increasing sample size substantially more data. This is also in line with our expectation as this makes the discrimination between signal and noise more difficult.

To summarize the results of our simulation study, `QTree` is sensitive to weaker dependence, but much more sensitive to the tail behavior of innovations/noise distributions.

We conclude that `QTree` works for sufficiently many observations very well even for limited data, across different dependence structures in the tree, and different distributions at the nodes.

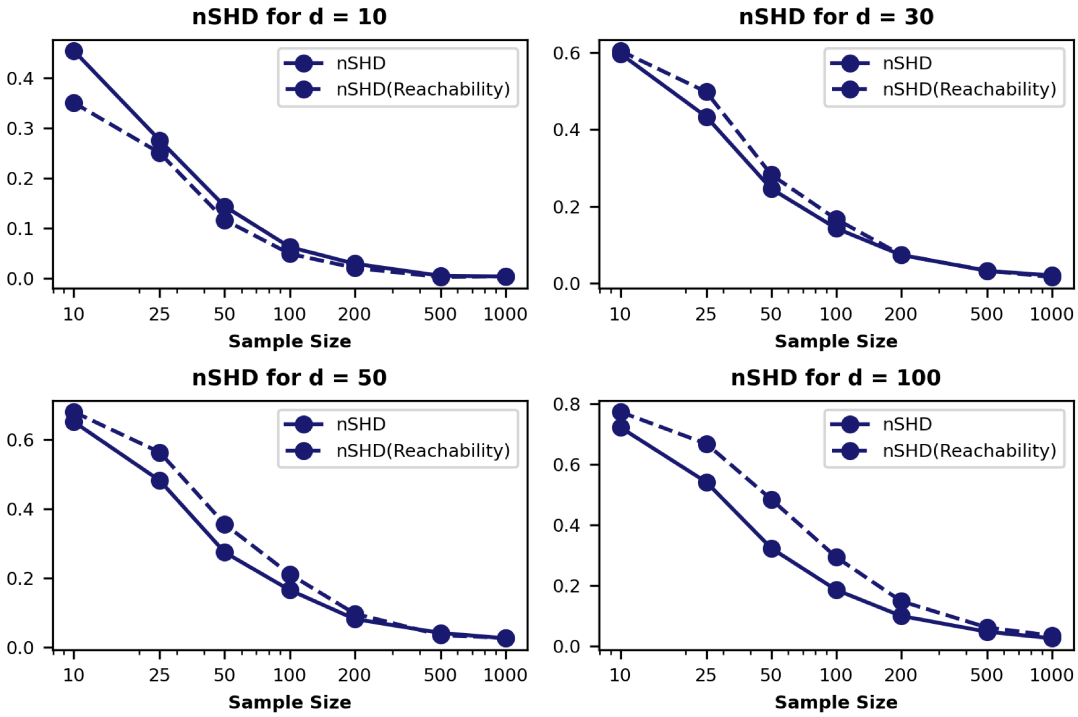

Fig. 12. Mean nSHD for the standard Gumbel setting (1) and graph size $d = 10$ (top left), $d = 30$ (top right), $d = 50$ (bottom left), $d = 100$ (bottom right) and noise-to-signal ratio $k = 30\%$. Solid lines denote the metric for the pair $(\mathcal{T}, \hat{\mathcal{T}})$ while dashed lines denote the metric for its reachability pair $(\mathcal{R}, \hat{\mathcal{R}})$. For all graph sizes, the nSHD quickly converges to 0 as n increases. Increasing the graph size only moderately decreases the performance.

6. Summary

In this paper, we proposed `auto-tuned QTree`, a new algorithmic solution to the Extremal River Problem—a benchmark problem for causal inference in extremes— combining the benefits of a new score matrix as input to Chu-Liu/Edmonds' algorithm with a stabilizing subsampling procedure. We also presented three new data sets of the Lower Colorado network for the Extremal River Problem, which are more challenging than the by now classic Upper Danube network data due to a large fraction of missing data and close spatial proximity between nodes. Across all four data sets, `auto-tuned QTree` performed very well, indeed better than previous state-of-the-art results. Our plug-and-play `Python` implementation in Tran (2021) can fit `QTree` on ten thousand observations in the range of 10 to 30 nodes on a personal laptop within half an hour. We proved that for a max-linear Bayesian network with Gumbel-Gaussian distributions for innovations and noise, the tree outputs of `QTree` are strongly consistent as the number of observations tends to infinity. Open research directions include (i) generalizations to learning directed acyclic graphs, (ii) better subsampling procedures with theoretical guarantees, and (iii) have the algorithm output a distribution over possible root-directed trees instead of a single best tree.

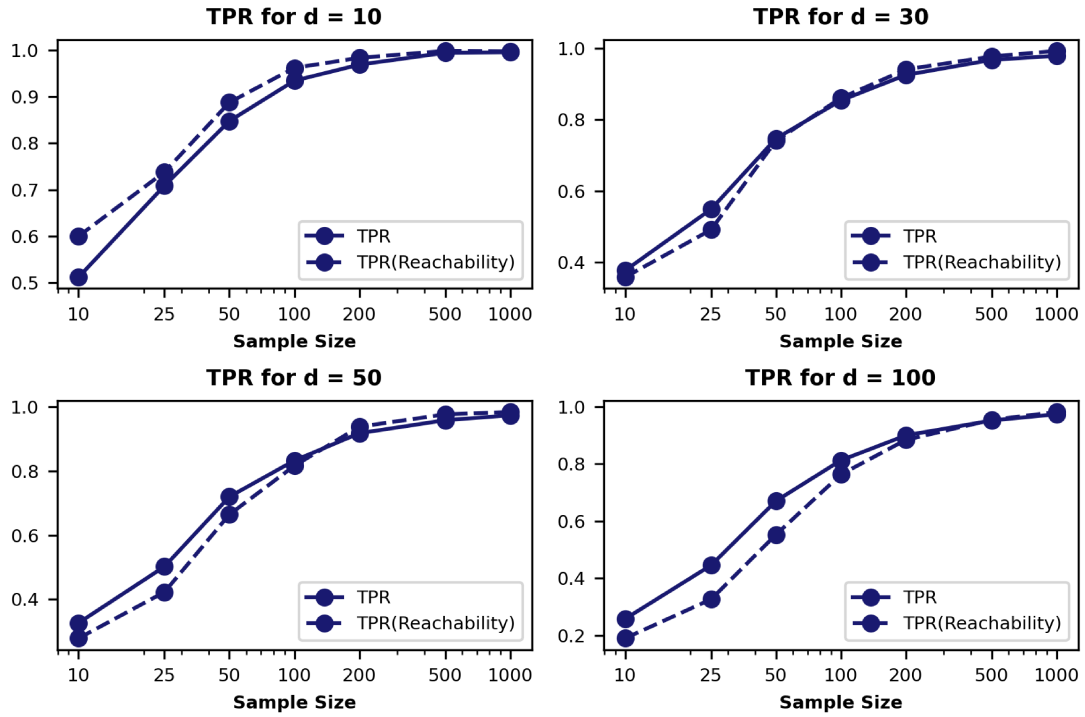

Fig. 13. Mean TPR for the standard Gumbel setting (1) and graph size $d = 10$ (top left), $d = 30$ (top right), $d = 50$ (bottom left), $d = 100$ (bottom right) and noise-to-signal ratio $k = 30\%$. Solid lines denote the metric for the pair $(\mathcal{T}, \hat{\mathcal{T}})$ while dashed lines denote the metric for its reachability pair $(\mathcal{R}, \hat{\mathcal{R}})$. For all graph sizes, the TPR quickly converges to 1 as n increases. Increasing the graph size only moderately decreases the performance.

Acknowledgements

We thank the Lower Colorado River Authority for providing the original data. We are also grateful to Sebastian Engelke for providing Figure 3 of the Upper Danube Basin as well as the declustered data, now available at Engelke et al. (2019). Ngoc Tran gratefully acknowledges support by the Hausdorff Center for Mathematics and the University of Bonn, Germany.

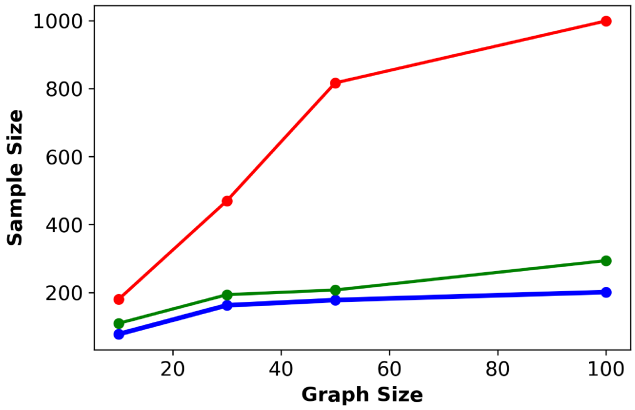

Fig. 14. Minimum amount of observations needed to reach a mean nSHD of 10%. Blue line for the standard Gumbel setting (1), green line for the weak dependence setting (2), and red line for the mixed distribution setting (3). For the standard Gumbel setting (1) and its weak dependence version (2), increasing the graph size d, the amount of data needed only increases moderately to reach the same level of performance. The mixed distribution setting (2) requires for increasing sample size substantially more data.

Supplementary Material

S1. Proof of the complexity of `QTree`

We work with the solution of the max-linear Bayesian network on a tree $\mathcal{T} = (V, E)$ defined in eq. (3) of the paper (Baccelli et al., 1992, §3); (Gissibl and Klüppelberg, 2018, Theorem 2.2). Let $C^* = (c^*_{ij})$ be the matrix of longest paths, also known as the *Kleene star of* $C = (c_{ij})$. Then

$$X_i = \bigvee_{j : j \rightsquigarrow i \in \mathcal{T}} (c^*_{ij} + Z_j), \quad c^*_{ij}, Z_{ij} \in \mathbb{R}, \quad i \in V. \tag{S1}$$

If there is no path $j \rightsquigarrow i$, then, by definition, $c^*_{ij} := -\infty$.

Lemma S1 concerns the noise-free case, and Lemma S2 concerns the noisy case.

Lemma S1. *Let* $\mathcal{X} = \{x^1, \dots, x^n\}$ *be i.i.d observations from the max-linear model given by* (S1), not *corrupted with noise. Assume that the* Z_i *are independent and have continuous distributions. Define*

$$\hat{c}_{ij} = \min_{x \in \mathcal{X}} (x_i - x_j). \tag{S2}$$

Suppose that for each edge $j \to i$ *such that* $c_{ij} > -\infty$, *there exist at least* two *observations* $x \in \mathcal{X}$ *where* j *causes* i. *Then* C^* *can be uniquely recovered from* $\hat{C}$ *since*

$$\hat{c}_{ij} = c^*_{ij} \iff \textit{min in (S2) is achieved at least twice.} \tag{S3}$$

In particular, C^* can be computed in time $O(|V|^2 n)$. If for every node, each of the parents is independently and equally likely to be the one that achieves the maximum, then the matrix C^* is recovered exactly for $n = O(|V|(\log(|V|))^2)$.

Proof. Since a root-directed spanning tree has at most one path between any pair of nodes (j, i) we have for an edge $j \to i$ that $c^*_{ij} = c_{ij}$. Furthermore, as indicated at the beginning of Section 2.2 in the Paper, if for an observation x the value at j causes that at i, then $x_i = c^*_{ij} + x_j$. If j does not cause i, then $x_i > c^*_{ij} + x_j$. Rearranging motivates the estimator (S2) with properties as stated in (Gissibl et al., 2021, Proposition 1). Equation (S3) follows from (Gissibl et al., 2021, Lemma 1).

Now we prove the complexity claim. Since there are $O(|V|^2)$ many edges, and for each edge we need $O(n)$ operations to compute the minimum in (S2), the complexity is $O(|V|^2 n)$. The number of observations needed, so that each edge is seen at least twice, is a variant of coupon-collecting (Boneh and Hofri, 1997; Boneh and Papanicolaou, 1996), where each node must collect two coupons (parents) among its set of parents. Since the nodes are collecting the coupons simultaneously, by the union bound, the number of observations needed is at most $\log(|V|)$ times the number of observations needed for the node with highest degree to collect all of its coupons, which in turn is $O(|V| \log(|V|))$. □

Lemma S2 (Complexity of QTree)**.** *`QTree` Algorithm 1 runs in time $O(|V|^2 n)$.*

Proof. For each pair $i, j \in V, i \neq j$, to estimate w_{ij}, one needs to compute the α-th quantile of $\mathcal{X}_j$, the $\underline{r}$-th quantile and the mean of $\mathcal{X}_{ij}(\alpha)$. Since α and $\underline{r}$ are fixed in advance, the empirical quantiles can be computed in time $O(n)$, see Musser (1997). As there are $O(|V|^2)$ pairs, computing $W = (w_{ij})$ takes $O(|V|^2 n)$. Chu–Liu/Edmonds' algorithm runs on the complete bidirected graph supported by W, and thus takes $O(|V|^2)$, see Gabow et al. (1986). So the complexity of QTree is $O(|V|^2 n + |V|^2) = O(|V|^2 n)$. □

S2. Proof of the Consistency Theorem

In this section, we prove Theorem 1 of the Paper, which we recall here for ease of reference.

> **Gumbel-Gaussian noise model.** For $i \in V$, the innovations Z_i are i.i.d. Gumbel$(\beta, 0)$ (location 0 and scale β), the independent noise variables ε_i are i.i.d with symmetric, light-tailed density f_ε satisfying
>
> $$f_\varepsilon(x) \sim e^{-Kx^p} \text{ as } x \to \infty, \tag{S4}$$
>
> for some $p > 1$ and $\gamma, K > 0$ and the derivative of f_ε exists in the tail region. Throughout, for two functions a, b, positive in their right tails, we write $a(x) \sim b(x)$ as $x \to \infty$ for $\lim_{x\to\infty} a(x)/b(x) = c$, where $c > 0$ is some arbitrary constant.

Theorem S1 (Theorem 1 of the Paper)**.** *Assume the Gumbel-Gaussian noise model. (a) There exists an $r^* > 0$ such that for any pair $0 < \underline{r} < \overline{r} < r^*$, the `QTree` algorithm with score matrix $W = (w_{ij})$ defined as the lower quantile gap*

$$w_{ij}(\underline{r}, \overline{r}) := \frac{1}{n_{ij}} \left(Q_{\mathcal{X}_{ij}(\alpha)}(\overline{r}) - Q_{\mathcal{X}_{ij}(\alpha)}(\underline{r})\right)^2 \tag{S5}$$

returns a strongly consistent estimator for the tree $\mathcal{T}$ as the sample size $n \to \infty$.
(b) There exists an $r^* > 0$ such that for any $0 < \underline{r} < r^*$, the `QTree` *algorithm with score matrix $W = (w_{ij})$ defined as the quantile-to-mean gap*

$$w_{ij}(\underline{r}) := \frac{1}{n_{ij}} \left(\mathbb{E}(\mathcal{X}_{ij}(\alpha)) - Q_{\mathcal{X}_{ij}(\alpha)}(\underline{r}) \right)^2 \tag{S6}$$

returns a strongly consistent estimator for the tree $\mathcal{T}$ as the sample size $n \to \infty$.

The proof of this theorem comes in a series of steps. Moreover, for simplicity, we omit the normalization by n_{ij} and the squaring in (S5) and (S6) as this leaves the proof unchanged. Also we set in the proof $\alpha = 0$.

As a preliminary result, Lemma S3 identifies a set of 'good' deterministic input matrices $W = (w_{ij})$, where if we apply the `QTree` algorithm to such an input, then it returns the true tree $\mathcal{T}$ *exactly*. The proof then reduces to the problem of proving that as $n \to \infty$, the matrices W_n derived from data converge a.s. to a 'good' W. Intuitively, W is 'good' if for each node j, the weight w_{ij} is smallest when i is the child of j. For the root we have a special explicit condition. For each fixed j, we split the set of node pairs $\{(j, i) : j, i \in V, i \neq j\}$ into three scenarios:

- $j \rightsquigarrow i$, that is, i is a descendant j in the true tree,
- $i \rightsquigarrow j$, that is, i is an ancestor of j in the true tree, and
- $i \not\rightsquigarrow j$, that is, i is neither of the above.

We first consider the case where $W = (w_{ij})$ is the matrix of lower quantile gaps (S5) of the *true* distribution. Note that this W is no longer random. The goal is to show that if the true quantiles are known, then one can choose the parameters $(\underline{r}, \overline{r})$ such that W is good.

Next Proposition S1 gives an explicit representation for w_{ij} in each of the three scenarios above as the lower quantile gap of a certain family of distributions $(F^b : b \in \mathbb{R})$, parametrized by a *single parameter* b, one value for each edge $j \to i$. Then, we use a calculus of variation argument to detail how w_{ij} changes as b varies. This allows us to show (cf. Corollary S1 and Lemma S6) that among the three scenarios above, there exist some choices of quantile levels $(\underline{r}, \overline{r})$ such that for *any* fixed j, w_{ij} is smallest when i is the child of j in the true tree. A separate argument is made for the root. Thus, this proves that if the true quantiles are known, then the resulting W is good.

Finally, we invoke the fact that the empirical quantiles converge a.s. to the true quantiles as $n \to \infty$, and thus the empirical w_{ij} are a.s. close to the true ones. A union bound over the d nodes of the graph thus says that, the empirical W_n is a.s. 'good' as $n \to \infty$, and thus proves the Consistency Theorem for the lower quantile gap.

The proof for the quantile-to-mean gap is similar, with Proposition S2 playing the role of Proposition S1.

Lemma S3 (A criterion for 'good' inputs W)**.** *Let $W = (w_{ij})$ be a score matrix such that each true edge $j \to i \in \mathcal{T}$ satisfies*

$$w_{ij} < w_{i'j} \text{ for all } i' \in V, i' \neq i, j, \tag{S7}$$

and in addition, the true root i^* satisfies

$$\min_{i'} w_{i'i^*} > \max_{i,j:j\to i} w_{ij}. \tag{S8}$$

Then the `QTree` *algorithm applied to input W returns the true tree $\mathcal{T}$.*

Proof. `QTree` applies Chu–Liu/Edmonds' algorithm to find a minimum directed spanning tree from the complete graph with score matrix W, and returns that tree. We shall prove that under the conditions (S7) and (S8) on W, Chu–Liu/Edmonds' algorithm would converge after one iteration and returns the true tree $\mathcal{T}$. Indeed, let $\mathcal{G}$ denote the graph that consists of the smallest outgoing edge at each node. By (S7), $\mathcal{G} = \mathcal{T} \cup i^* \to i'$ for some node $i' \in V$. By Chu–Liu/Edmonds' algorithm, the minimum spanning tree $\mathcal{T}_w$ is a subset of $\mathcal{G}$. In particular, $\mathcal{T}_w$ is a minimum spanning tree of $\mathcal{G}$. By (S8), edge $i^* \to i'$ is the maximal edge. Since it belongs to the unique cycle in $\mathcal{G}$, deleting this edge would yield the minimum directed spanning tree of $\mathcal{G}$. Therefore $\mathcal{T}_w = \mathcal{T}$. □

S2.1. Proof of Theorem 1 for the lower quantile gap

S2.1.1. For known quantiles, W is 'good' for appropriate choices of $(\underline{r}, \overline{r})$

In this subsection we work with the lower quantile gap matrix $W = (w_{ij})$ derived from the *true* quantiles of the distributions of $X_i - X_j$ under the Gumbel-Gaussian model, for some quantile levels $(\underline{r}, \overline{r})$. The goal is to show that there exist some appropriate choices of $(\underline{r}, \overline{r})$ such that the resulting W is 'good', that is, it satisfies Lemma S3.

The first main result is Proposition S1, which gives an explicit representation for w_{ij} in the three scenarios. We start with the necessary definitions to state it.

Recall the definition of C^* from the beginning of Section 1. Since the true graph is a tree, if $j \rightsquigarrow i$, there is a unique directed path from j to i. Let $\bar{c}_{ij}$ denote the sum of all the edges along this unique path. Path uniqueness implies that $\bar{c}_{ij} = c^*_{ij}$ and C^* is transitive, i.e. $c^*_{ij} = c^*_{ik} + c^*_{kj}$ if $j \rightsquigarrow k \rightsquigarrow i$. Thus, by the Helmholtz decomposition on graphs (Lim, 2015, equation 2.6), c^*_{ij} is an edge flow. That is, there exists a unique $t^* \in \mathbb{R}^d$ with $t^*_1 = 0$ such that for all $j \to i \in \mathcal{G}$,

$$c^*_{ij} = t^*_i - t^*_j. \tag{S9}$$

For each $i \in V$, define the constant

$$\theta_i := \sum_{k \rightsquigarrow i} \exp(-t^*_k/\beta). \tag{S10}$$

For $b \in \mathbb{R} \cup \{-\infty\}$, define the random variable

$$\xi_b := (\varepsilon_i - \varepsilon_j) + ((Z_i - Z_j) \vee b) \tag{S11}$$

with the convention that $\xi_{-\infty} := (\varepsilon_i - \varepsilon_j) + (Z_i - Z_j)$. Let F^b denote the distribution function of ξ_b and $q_r(F^b)$ the r-th quantile of F^b for $r \in (0, 1)$. These quantities are deterministic and do not depend on i, j since by assumption, $\varepsilon_i, \varepsilon_j$ are i.i.d and Z_i, Z_j are i.i.d.

Proposition S1. *Assume the Gumbel-Gaussian model. Fix* $0 < \underline{r} < \overline{r} < 1$. *Let* $w_{ij} = w_{ij}(\underline{r}, \overline{r})$ *be the lower quantile gap* (S5). *Fix* $j \in V$. *For* $i \in V, i \neq j$, *we have three cases.*

(1) *If* $j \rightsquigarrow i$, *then* $w_{ij} = q_{\overline{r}}(F^b) - q_{\underline{r}}(F^b)$ *for* $b = \beta(\log\theta_j - \log(\theta_i - \theta_j))$.
(2) *If* $j \not\rightsquigarrow i$, *then* $w_{ij} = q_{\overline{r}}(F^b) - q_{\underline{r}}(F^b)$ *for* $b = -\infty$.
(3) *If* $i \rightsquigarrow j$, *then* $w_{ij} = q_{1-\underline{r}}(F^b) - q_{1-\overline{r}}(F^b)$ *for* $b = \beta(\log\theta_i - \log(\theta_j - \theta_i))$.

Proof. We first consider the noise-free case. Observe that by (S11) ξ_b simplifies to $(Z_i - Z_j) \vee b$. Therefore, it is sufficient to prove that w_{ij} equals the lower quantile gap of $(Z_i - Z_j) \vee b$. For $i \in V$, let $\bar{X}_i := X_i - t_i^*$. Then $\bar{X}_i - \bar{X}_j$ is a constant translation of $X_i - X_j$, so the lower quantile gap of the two corresponding distributions are the same. In other words, it is sufficient to prove the Proposition for $\bar{X}$ instead of X. Let $\bar{Z}_i := Z_i - t_i^*$. Then

$$\begin{aligned}
\bar{X}_i = X_i - t_i^* &= \bigvee_{j:j\rightsquigarrow i} (c_{ij}^* + Z_j) - t_i^* && \text{by (S1)} \\
&= \bigvee_{j:j\rightsquigarrow i} (t_i^* - t_j^* + Z_j) - t_i^* && \text{by (S9)} \\
&= \bigvee_{j:j\rightsquigarrow i} \bar{Z}_j. && \text{(S12)}
\end{aligned}$$

For each ordered pair (i, j), define

$$S_i = \bar{Z}_i \vee \bigvee_{i'\neq i, i'\rightsquigarrow i, i'\not\rightsquigarrow j} \bar{Z}_{i'} \qquad\qquad S_j = \bar{Z}_j \vee \bigvee_{j'\neq j, j'\rightsquigarrow j} \bar{Z}_{i'} \qquad \text{(S13)}$$

In Figure S1 we illustrate the two index sets of the random variables S_i and S_j.

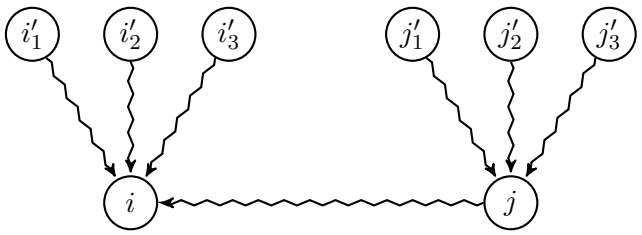

Fig. S1. Illustration of the index sets of S_i and S_j for an ordered pair (i, j). The index set for S_i includes besides i also $i'_1, \ldots, i'_3$ and all nodes on the paths $j \rightsquigarrow i$ (excluding j), $i'_k \rightsquigarrow i$ for $k = 1, \ldots, 3$, while the index set for S_j includes $j, j'_1, \ldots, j'_3$ and all nodes on the paths $j'_k \rightsquigarrow j$ for $k = 1, \ldots, 3$.

By definition, S_i and S_j are independent. Since $\bar{Z}_i$'s are translated independent Gumbel(β) by assumption, standard properties of the Gumbel(β) distribution yield that S_i and S_j are also translated independent Gumbel(β). The exact constants of translation depend on the relation between i and j, as this dictates the definition of S_i and S_j. Now we consider the three cases. In the first case, $j \rightsquigarrow i$. Then, (S12) implies $\bar{X}_i = S_i \vee S_j$ and $\bar{X}_j = S_j$.

A short computation yields $S_i \overset{d}{=} Z_i + \beta \log(\theta_i - \theta_j)$, $S_j \overset{d}{=} Z_j + \beta \log \theta_j$. Therefore, denoting $\overset{\mathrm{d}}{=}$ equality in distribution,

$$\begin{aligned}
\bar{X}_i - \bar{X}_j &= (S_i \vee S_j) - S_j = (S_i - S_j) \vee 0 \\
&\overset{d}{=} (Z_i - Z_j - \beta(\log \theta_j - \log(\theta_i - \theta_j))) \vee 0 \\
&= ((Z_i - Z_j) \vee \beta(\log \theta_j - \log(\theta_i - \theta_j))) - \beta(\log \theta_j - \log(\theta_i - \theta_j)) \\
&= ((Z_i - Z_j) \vee b) - \beta(\log \theta_j - \log(\theta_i - \theta_j)),
\end{aligned}$$

where $b = \beta(\log \theta_j - \log(\theta_i - \theta_j))$. Since $\beta(\log \theta_j - \log(\theta_i - \theta_j))$ is a translation constant, the quantile gap of $\bar{X}_i - \bar{X}_j$ is equal to the quantile gap of $(Z_i - Z_j) \vee b$. This concludes the case $j \rightsquigarrow i$. Computations for the third case, $i \rightsquigarrow j$, is similar, with the role of i and j reversed, $\underline{r}$ is replaced by $1 - \overline{r}$, and $\overline{r}$ is replaced by $1 - \underline{r}$. For the second case, $i \not\sim j$, then $\bar{X}_i = S_i, \bar{X}_j = S_j$, where $S_j \overset{d}{=} \beta \log \theta_j + Z_j$ and $S_i \overset{d}{=} \beta \log \theta_i + Z_i$. Then

$$\bar{X}_i - \bar{X}_j = S_i - S_j \overset{d}{=} Z_i - Z_j + \beta(\log \theta_i - \log \theta_j).$$

Since $\beta(\log \theta_i - \log \theta_j)$ is a translation constant, the quantile gap of $\bar{X}_i - \bar{X}_j$ is equal to the quantile gap of $Z_i - Z_j$, as claimed. □

S2.1.2. How the lower quantile gap w_{ij} varies with b

Now, we aim to show through a variational argument that under the Gumbel-Gaussian assumption, among the three scenarios of Proposition S1, w_{ij} is smallest when it falls in a subset of case (1), namely, $j \to i$. We first give an overview. By Proposition S1, the lower quantile gaps w_{ij} in cases (1) and (2) are all of the form $q(b, \overline{r}) - q(b, \underline{r})$ for some constant $b = b(i, j)$. In particular, for fixed j, $b(i, j)$ is largest when $j \to i$. Lemma S5 says that one can choose the quantile levels $(\underline{r}, \overline{r})$ such that $q(b, \overline{r}) - q(b, \underline{r})$ is monotone *increasing* as a function of b on a large interval. Corollary S1 then shows that a good choice can be made so that for each fixed j, the quantile gap is smallest for the edge from j to its child $ch(j)$. Case (3) of Proposition S1, where i is an ancestor of j, is handled by Lemma S6. The Gumbel-Gaussian assumption comes in through Lemma S4, which is a technical result that gives an explicit form for the density of the noise differences $\eta := \varepsilon_i - \varepsilon_j$. Intuitively, it shows that under the Gumbel-Gaussian model, the tail of η is lighter than the tail of the signal differences $Z_i - Z_j$. This is a key observation exploited in the proofs.

Lemma S4. *Under the Gumbel-Gaussian model, for any pair of nodes $i, j \in V, i \neq j$, $\xi := Z_i - Z_j$ has density*

$$f_\xi(x) = \frac{e^{x/\beta}}{\beta(1 + e^{x/\beta})^2} \sim \frac{1}{\beta} e^{-x/\beta} \text{ as } x \to \infty, \tag{S14}$$

and $\eta := \varepsilon_i - \varepsilon_j$ has density

$$f_\eta(x) \sim x^{1-p/2}\, e^{-Kx^p} \text{ as } x \to \infty. \tag{S15}$$

Proof. Computing the convolution integral yields

$$\mathbb{P}(Z_i - Z_j > x) = \frac{1}{1 + e^{x/\beta}}, \quad x \in \mathbb{R}, \tag{S16}$$

and taking the derivative gives the first statement. For the second statement, the density f_ε is a density with Gaussian tail in the sense of Balkema et al. (1993):

$$f(x) \sim \gamma(x) e^{-\psi(x)} \text{ as } x \to \infty,$$

for constants γ and $\psi(x) = Kx^p$. The asymptotic form of f_η follows by Laplace's integration principle as shown in (Balkema et al., 1993, page 2). □

Since f_ε is differentiable in the tail, f_η is also differentiable in the tail, and differentiation of (S15) yields the following formula for the derivative:

$$f'_\eta(x) \sim f_\eta(x)\Big(- Kpx^{p-1} + (1 - p/2)x^{-1}\Big) \tag{S17}$$

For functions with two arguments, let ∂_1 denotes the derivative in the first argument, ∂_2 denotes the derivative in the second argument, $\partial^2_{12} := \partial_1\partial_2$ denote the mixed second derivatives and so forth. Define the functions $H : \mathbb{R} \times \mathbb{R} \to [0, 1]$, $q : \mathbb{R} \times [0, 1] \to \mathbb{R}$ by

$$H(b, a) = P(\xi_b \le a), \quad q(b, r) = r\text{-th quantile of } \xi_b.$$

Lemma S5. *Under the Gumbel-Gaussian model, for each finite constant B, there exists some $r^* = r^*(B) \in (0, 1)$ such that*

$$\partial^2_{12} q(b, r) < 0 \quad \text{for all } r \in (0, r^*), b \le B.$$

Equivalently, for any pair $(\underline{r}, \overline{r})$ such that $0 < \underline{r} < \overline{r} < r^*$ and any pair (b', b) such that $b' < b \le B$,

$$q(b, \overline{r}) - q(b, \underline{r}) < q(b', \overline{r}) - q(b', \underline{r}). \tag{S18}$$

Proof. By definition,

$$H(b, q(b, r)) = r. \tag{S19}$$

We take derivatives of both sides, first with respect to r, then to b. Note that functions and derivatives of H are always evaluated at $(b, q(b, r))$ while those of q are evaluated at (b, r), so we suppress them in the notations. Differentiate both sides sof (S19) with respect to r gives

$$\partial_2 H \cdot \partial_2 q = 1. \tag{S20}$$

Now, differentiating both sides of (S19) with respect to b, we get

$$\frac{\partial}{\partial b} H_1(b, q(b, r)) = \partial_1 H + \partial_2 H \cdot \partial_1 q = 0,$$

therefore,

$$\partial_1 q = \frac{-\partial_1 H}{\partial_2 H}. \tag{S21}$$

Differentiate (S20) with respect to b using implicit differentiation and chain rules, we get

$$\begin{aligned} 0 = \frac{\partial}{\partial b}(\partial_2 H \cdot \partial_2 q) &= \frac{\partial}{\partial b}(\partial_2 H(b, q(b,r)) \cdot \partial_2 q + \partial_2 H \cdot \partial^2_{12} q \\ &= (\partial^2_{12} H + \partial^2_{22} H \cdot \partial_1 q) \cdot \partial_2 q + \partial_2 H \cdot \partial^2_{12} q \\ &= \frac{\partial^2_{12} H - \partial^2_{22} H \cdot \frac{\partial_1 H}{\partial_2 H}}{\partial_2 H} + \partial_2 H \cdot \partial^2_{12} q \quad \text{by (S20) and (S21).} \end{aligned} \tag{S22}$$

Rearranging the last equation gives

$$\partial^2_{12} q = \frac{\partial^2_{22} H \cdot \partial_1 H - \partial^2_{12} H \cdot \partial_2 H}{(\partial_2 H)^3}. \tag{S23}$$

For fixed b, by definition of H, $\partial_2 H$ is the density of ξ_b, so $\partial_2 H > 0$. So $\partial^2_{12} q(b,r) < 0$ if and only if

$$(\partial^2_{22} H \partial_1 H - \partial^2_{12} H \partial_2 H)(b, q(b,r)) < 0. \tag{S24}$$

Now we compute each of the terms $\partial_2 H, \partial_1 H, \partial^2_{12} H$ and $\partial^2_{22} H$ in the LHS of (S24) explicitly in terms of the density f_η of the noise difference $\eta = \varepsilon_i - \varepsilon_j$. Note that $\xi_b = \eta + (\xi \vee b)$ where $\xi := Z_i - Z_j$. Then we have for $\varepsilon > 0$ (see Fig. (a))

$$H(b+\varepsilon, a) - H(b,a) = \mathbb{P}(\eta + \xi \vee (b+\varepsilon) \le a) - \mathbb{P}(\eta + \xi \vee b \le a)$$

$$= \begin{cases} 0 & \text{if } \xi > b + \varepsilon, \\ \mathbb{P}(\eta + b + \varepsilon \le a) - \mathbb{P}(\eta + b \le a) & \text{if } \xi \le b \quad \text{(light shaded)}, \\ \mathbb{P}(\eta + b + \varepsilon \le a) - \mathbb{P}(\eta + \xi \le a) & \text{if } b \le \xi \le b + \varepsilon \quad \text{(dark shaded)}. \end{cases}$$

Since η and ξ are independent, this implies

$$H(b+\varepsilon, a) - H(b,a) = -\mathbb{P}(\xi \le b)\mathbb{P}(a - b - \varepsilon \le \eta \le a - b) + O(\varepsilon^2). \tag{S25}$$

Hence,

$$\partial_1 H(b,a) = \lim_{\varepsilon \downarrow 0} \frac{H(b+\varepsilon, a) - H(b,a)}{\varepsilon} = -\mathbb{P}(\xi \le b) f_\eta(a-b). \tag{S26}$$

A similar calculation gives (see Fig. (b))

$$\begin{aligned} \partial_2 H(b,a) = \lim_{\varepsilon \downarrow 0} \frac{H(b, a+\varepsilon) - H(b,a)}{\varepsilon} &= \mathbb{P}(\xi \le b) f_\eta(a-b) + \int_b^\infty f_\eta(a-x) f_\xi(x)\, dx \\ &=: (-\partial_1 H + A)(b,a). \end{aligned} \tag{S27}$$

Thus,

$$\begin{aligned} \partial^2_{12} H(b,a) &= -\mathbb{P}(\xi \le b) f'_\eta(a-b), \\ \partial^2_{22} H(b,a) &= \mathbb{P}(\xi \le b) f'_\eta(a-b) + \int_b^\infty f'_\eta(a-x) f_\xi(x)\, dx \\ &= -\partial^2_{12} H(b,a) + \int_b^\infty f'_\eta(a-x) f_\xi(x)\, dx. \\ &= (-\partial^2_{12} H + A_a)(b,a). \end{aligned}$$

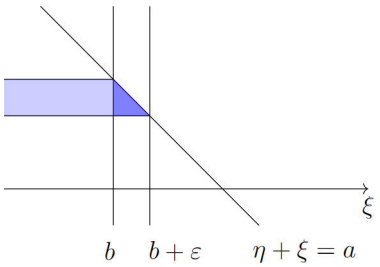

Fig. S2. $H(b+\varepsilon,a)-H(b,a)$ is the probability that (ξ,η) lies in the shaded regions (light + dark shaded)

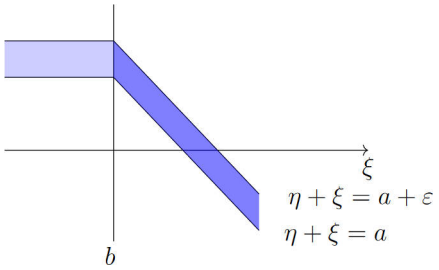

Fig. S3. $H(b,a+\varepsilon)-H(b,a)$ is the probability that (ξ,η) lies in the shaded region (light + dark shaded)

Now we have

$$\begin{aligned}&(\partial_{22}^2 H\partial_1 H-\partial_{12}^2 H\partial_2 H)(b,a)=(A_a\partial_1 H-A\partial_{12}^2 H)(b,a)\\ =&\mathbb{P}(\xi\le b)\left(-f_\eta(a-b)\int_b^\infty f_\eta'(a-x)f_\xi(x)\,dx+f_\eta'(a-b)\int_b^\infty f_\eta(a-x)f_\xi(x)\,dx\right).\end{aligned}$$

Thus, (S24) holds if and only if

$$-f_\eta(q(b,r)-b)\int_b^\infty f_\eta'(q(b,r)-x)f_\xi(x)\,dx+f_\eta'(q(b,r)-b)\int_b^\infty f_\eta(q(b,r)-x)f_\xi(x)\,dx<0. \tag{S28}$$

Now we need to show that for each constant B, there exists an $r^*(B)>0$ such that for all $r<r^*$ and $b\le B$, (S28) holds. Fix the constant B. Since the noise ε has no upper bound, η and hence ξ_b have unbounded support below. For each fixed b, $q(b,r)\to-\infty$ as $r\downarrow 0$. Therefore, there exists some sufficiently small $r^*>0$ such that for all $r<r^*$, $q(b,r)-b$ is a large negative number. Fix such an r^*. Therefore, for all $x>b$, $q(b,r)-x$ is a large negative number. This allows us to use Lemma S4 to make the LHS of (S28) explicit. In particular, by (S15), as $t\to\infty$,

$$f_\eta(t)=K_1 t^{1-p/2}e^{-Kt^p}(1+o(1)),\quad f_\eta'(t)=f_\eta(t)\Big(-Kpt^{p-1}+(1-p/2)t^{-1}\Big)(1+o(1)). \tag{S29}$$

Setting $t=|x-q(b,r)|$ and use the fact that f_η is symmetric, $f_\eta(q(b,r)-x)=f_\eta(|x-q(b,r)|)$, we have

$$\begin{aligned}&-f_\eta(q(b,r)-b)\int_b^\infty f_\eta'(q(b,r)-x)f_\xi(x)\,dx+f_\eta'(q(b,r)-b)\int_b^\infty f_\eta(q(b,r)-x)f_\xi(x)\,dx\\ =&f_\eta(q(b,r)-b)\int_b^\infty[-Kp(x-q(b,r))^{p-1}+(1-p/2)(x-q(b,r))^{-1}]f_\eta(q(b,r)-x)f_\xi(x)\,dx\\ &+[Kp(b-q(b,r))^{p-1}-(1-p/2)(b-q(b,r))^{-1}]f_\eta(q(b,r)-b)\int_b^\infty f_\eta(q(b,r)-x)f_\xi(x)\,dx\\ =&f_\eta(q(b,r)-b)\int_b^\infty A(x)f_\eta(q(b,r)-x)f_\xi(x)\,dx,\end{aligned}$$

where

$$A(x)=\Big(Kp[(b-q(b,r))^{p-1}-(x-q(b,r))^{p-1}]-(1-p/2)[(b-q(b,r))^{-1}-(x-q(b,r))^{-1}]\Big).$$

If $p \in (1,2]$, since $x > b$, the first term $(b - q(b,r))^{p-1} - (x - q(b,r))^{p-1}$ and the second term $-(1-p/2)[(b-q(b,r))^{-1} - (x-q(b,r))^{-1}]$ are both negative. If $p > 2$, since $x > b \gg q(b,r)$ the first term $(b - q(b,r))^{p-1} - (x - q(b,r))^{p-1}$ is a large negative number. Since $q(b,r)$ is a large negative number, $b - q(b,r)$ is a large positive number, so $(b-q(b,r))^{-1}, (x-q(b,r))^{p-1} < 1$. Thus $|(1-p/2)[(b-q(b,r))^{-1} - (x-q(b,r))^{-1}]| \leq |1-p/2|$. For this reason, $A(x) < 0$ for all $p > 1$, $x > b$, while $f_\eta, f_\xi > 0$ everywhere as they are densities. Thus the integral is negative, that is, (S28) holds for all $r \in (0, r^*)$ and $b \leq B$, as needed. □

Below we denote $ch(j)$ the child of node j.

Corollary S1. *Under the Gumbel-Gaussian model, there exists an $r_1^* > 0$ such that: for all $0 < \underline{r} < \overline{r} < r_1^*$, for all $j \in V$ and for all $i' \in V, i' \neq j, ch(j)$ and either $j \rightsquigarrow i'$ or $j \not\rightsquigarrow i'$, then*

$$w_{ch(j)\,j} < w_{i'j}.$$

Proof. It is sufficient to show that the above holds with some constant $r^*(j)$ for each fixed j, then set $r_1^* = \min_j r^*(j)$. Fix j and i' as stated. Let $b^* := \beta(\log\theta_j - \log(\theta_{ch(j)} - \theta_j))$, and let $r^*(j)$ be the constant r^* that works for $B = b^*$ in Lemma S5. By Proposition S1,

$$w_{ch(j)\,j} = q(b^*, \overline{r}) - q(b^*, \underline{r}).$$

Now we consider two cases.
Case 1: i' is a descendant of j, that is, $j \rightsquigarrow i'$. Then by Proposition S1,

$$w_{ji'} = q(b, \overline{r}) - q(b, \underline{r})$$

where $b = \beta(\log\theta_j - \log(\theta_{i'} - \theta_j))$. But since $i' \neq ch(j)$, i' must be a descendant of i as well. By definition of θ's in (S10), $i \rightsquigarrow i'$ implies $\theta_{i'} > \theta_i$. Therefore, $b < b^*$, so by (S18), $w_{ij} < w_{i'j}$. This concludes case 1.
Case 2: $j \not\rightsquigarrow i'$. Then by Proposition S1,

$$w_{i'j} = q(-\infty, \overline{r}) - q(-\infty, \underline{r}).$$

Since $-\infty < b^*$, so by (S18), $w_{ij} < w_{i'j}$. This concludes case 2. □

Lemma S6. *There exists some $r_2^* > 0$ such that for all $0 < \underline{r} < \overline{r} < r_2^*$, for all $j \in V$, $i' \rightsquigarrow j$ implies*

$$q(b, \overline{r}) - q(b, \underline{r}) < q(b', 1-\underline{r}) - q(b', 1-\overline{r}), \tag{S30}$$

where $b = \beta(\log\theta_j - \log(\theta_{ch(j)} - \theta_j))$ and $b' = \beta(\log\theta_{i'} - \log(\theta_j - \theta_{i'}))$. In particular, if $i' \rightsquigarrow j$, then for all quantile levels $\underline{r}, \overline{r}$ such that $0 < \underline{r} < \overline{r} < r_2^*$,

$$w_{ch(j)\,j} < w_{i'j}. \tag{S31}$$

Proof. It is sufficient to prove that (S30) holds for each fixed j with some constant $r_2^*(j)$, and then set $r_2^* = \min_j r_2^*(j)$. Fix j. First, we do some manipulations on (S30) to relate it to the partial derivatives of H. Define

$$\mathcal{B} := \{\beta(\log\theta_j - \log(\theta_i - \theta_j)) : i, j \in V, j \rightsquigarrow i\}. \tag{S32}$$

Note that (S30) is equivalent to

$$\partial_2 q(b,r) < \partial_2 q(b', 1-r) \quad \text{for all } r \in (0, r_2^*) \text{ and for all } b, b' \in \mathcal{B}. \tag{S33}$$

By (S20), we have

$$\partial_2 q(b,r) - \partial_2 q(b', 1-r) = \frac{1}{\partial_2 H(b, q(b,r))} - \frac{1}{\partial_2 H(b', q(b', 1-r))}.$$

By (S27), $\partial_2 H > 0$ point-wise, thus our goal now is to show that for sufficiently small r,

$$\partial_2 H(b', q(b', 1-r)) - \partial_2 H(b, q(b,r)) < 0 \tag{S34}$$

for all $b, b' \in \mathcal{B}$, that is, some finite set of constants.We shall do this by writing $\partial_2 H$ in terms of the tail densities f_η and f_ξ using (S27), then apply Lemma S4. Indeed, by (S27),

$$\partial_2 H(b', a) = \mathbb{P}(\xi \le b') f_\eta(a - b') + \int_{b'}^{\infty} f_\eta(a-x) f_\xi(x)\, dx$$

By Lemma S4, f_ξ has heavier tail than f_η, so for $a \to \infty$, the main contribution from $\int_{b'}^{\infty} f_\eta(a-x) f_\xi(x)\, dx$ comes from $f_\xi(a)$. That is, for large a, there exists some constant $b_1 > 0$ such that

$$\partial_2 H(b', a) > b_1 f_\xi(a). \tag{S35}$$

Now we consider $\partial_2 H(b, -a)$. From (S27),

$$\partial_2 H(b, -a) = \mathbb{P}(\xi \le b) f_\eta(-a - b) + \int_{b}^{\infty} f_\eta(-a-x) f_\xi(x)\, dx.$$

Again, for large a

$$f_\eta(-a - x) < f_\eta(-a - b) \text{ for all } x > b.$$

Therefore, we can bound the second term above as

$$\int_b^\infty f_\eta(-a-x) f_\xi(x)\, dx < f_\eta(-a-b) \int_b^\infty f_\xi(x)\, dx = f_\eta(-a-b)\mathbb{P}(\xi > b).$$

Adding in the first term, we get that for large a,

$$\partial_2 H(b, -a) < f_\eta(-a - b)$$

Combining this with (S35) and noting that $\partial_2 H(b, a)$ is just the density $f_{\xi_b}(a)$ of ξ_b, we get

$$f_{\xi_b}(-a) = O(f_{\xi_{b'}}(a)) \tag{S36}$$

for all $b, b' \in \mathcal{B}$ and a large. Now $\partial_2 H(b, q(b,r))$ is just the slope of the cdf of f_{ξ_b} at its r-th quantile. Therefore, for r small, by (S36), $\partial_2 H(b', q(b', 1-r)) < \partial_2 H(b, q(b,r))$ which proves (S34) and thus completes the proof of (S30). The last statement follows from Proposition S1, case (3). □

Corollary S2. *If the true quantiles are known, then there exist some choices of* $(\underline{r}, \overline{r})$ *such that the lower quantile gap matrix* W *satisfies the conditions of Lemma S3, that is,* (S7) *and* (S8).

Proof. Set $r^* = \min(r_1^*, r_2^*)$ where r_1^* comes from Corollary S1, and r_2^* comes from Lemma S6. Let $(\underline{r}, \overline{r})$ be any pair such that $0 < \underline{r} < \overline{r} < r^*$, and let W be the corresponding lower quantile gap matrix with the true quantiles. Then (S8) holds because of (S31) and the fact that for the root r of the root-directed spanning tree, $i' \rightsquigarrow r$ holds for every $i' \neq r$. Corollary S1 and Lemma S6 together guarantee that (S7) is satisfied for W. □

Proof of Theorem 1 for the lower quantile gap

Fix $(\underline{r}, \overline{r})$ such that Corollary S2 holds, and let W be the corresponding lower quantile gap matrix derived from the true quantiles. Let W_n be the lower quantile gap matrix derived from an empirical distribution with sample size n. Note that the set of 'good' matrices, that is, those that satisfy Lemma S3, is an open polyhedral cone in the space of matrices $\mathbb{R}^{d\times d}$, since the conditions of 'goodness' is a set of linear inequalities. By Corollary S2, W is a point inside this cone. Recall that empirical quantiles converge a.s. as $n \to \infty$ to the true ones for continuous limit distributions, hence, also the empirically-derived lower quantile gap converges a.s.. By a union bound over the $d^2 - d$ possible edge pairs (i, j), for any metric D (e.g. induced by a matrix norm), we thus have $D(W_n, W) \to 0$ a.s. The Consistency Theorem then follows from Lemma S3. □

S2.2. Proof of Theorem 1 for the quantile-to-mean gap

Our proof follows the same structure as the previous proof, but the calculations in all steps are a bit simpler, since there is only one quantile parameter to deal with. First, expectation is linear, so we work with empirical means $\bar{X}_i$ for $i \in V$ and mention in passing that they converge a.s. to the true mean as $n \to \infty$. The analogue of Proposition S1 is the following.

Proposition S2. *Fix* $\underline{r} \in [0, 1)$*, and let* w_{ij} *be the quantile-to-mean gap* (S6)*. Assume the Gumbel-Gaussian model. Then*

(1) *If* $j \rightsquigarrow i$*, then* $w_{ij} = -q_{\underline{r}}(\xi^b)$ *where* $b = \beta(\log\theta_j - \log(\theta_i - \theta_j))$.
(2) *If* $j \not\rightsquigarrow i$*, then* $w_{ij} = -q_{\underline{r}}(\xi^b)$ *where* $b = -\infty$.
(3) *If* $i \rightsquigarrow j$*, then* $w_{ij} = q_{1-\underline{r}}(\xi^b)$ *where* $b = \beta(\log\theta_i - \log(\theta_j - \theta_i))$.

Instead of a lengthy proof of the analog of Proposition S1 by duplicating arguments, we provide some informal reasoning. We check that our quantile-to-mean gaps w_{ij} satisfy the inequalities of Corollary S1 and Lemma S6 by first checking the noise-free case, where $\varepsilon_i \equiv \varepsilon_j \equiv 0$. We consider the three cases of Proposition S2.

(a) If $j \rightsquigarrow i$. Then ξ^b has a left-most atom at $b = \beta(\log\theta_j - \log(\theta_i - \theta_j))$, so for sufficiently small r, $w_{ij} = -b$. This is minimal when i is a direct descendant of j. So Corollary S1 for the case $j \rightsquigarrow i$ holds in the noise-free case.

(b) If $j \not\rightsquigarrow i$. Then ξ^b has no left-most atom, so as $\underline{r} \downarrow 0$, $q_{\underline{r}}(\xi^b) \to -\infty$, so $w_{ij} \to \infty$. So Corollary S1 also holds in the noise-free case for the remaining case, $j \not\rightsquigarrow i$.

(c) If $i \rightsquigarrow j$. Then ξ^b has a left-most atom, but no right-most atom. Again, as $\underline{r} \downarrow 0$, $q_{1-\underline{r}}(\xi^b) \to \infty$, so $w_{ij} \to \infty$. Thus, Lemma S6 holds in the noise-free case.

Now we consider the effect of noise. We send $\underline{r} \downarrow 0$. As long as $\eta := \varepsilon_i - \varepsilon_j$ has lighter tail than $Z_i - Z_j$, as guaranteed by Lemma S4, then we have the following.

- In case (1), $q_{\underline{r}}(\xi^b)$ is dominated by the lower tail of η.
- In case (2), $q_{\underline{r}}(\xi^b)$ is dominated by the lower tail of $Z_i - Z_j$ and, in particular, is going to $-\infty$ at a faster rate than case (1).
- In case (3), $q_{1-\underline{r}}(\xi^b)$ is dominated by the upper tail of $Z_i - Z_j$, and in particular, is going to ∞ at a faster rate than case (1).

This domination calculation is the same calculation done in the proof of Lemma S6. The above says that for fixed j, for small enough $\underline{r}$, the minimum of $\{w_{ij} : i \neq j, i \in V\}$ lies in case (1). Within case (1), we want to make sure that, if w_{ij} is smallest, then i is the child of j. Indeed, write

$$\xi^b = \varepsilon_i - \varepsilon_j + \xi'_{ij}$$

where $\xi'_{ij} = (Z_i - Z_j) \vee (\beta(\log\theta_j - \log(\theta_i - \theta_j)))$. For fixed j, $(\xi'_{ij} : j \rightsquigarrow i)$ is a particular family of distribution indexed by i. By a decoupling argument, it is sufficient to show that $q_{\underline{r}}(\xi'_{ij})$ is smallest when i is the child of j. But this reduces to the noise-free case, which we already proved above. This finishes the proof of Theorem 1 for the quantile-to-mean gap. □

S3. Supplemental Figures for Section 4.2

Below we present the figures analogous to Figure 10 of the Paper for the different data sets of the Colorado river network.

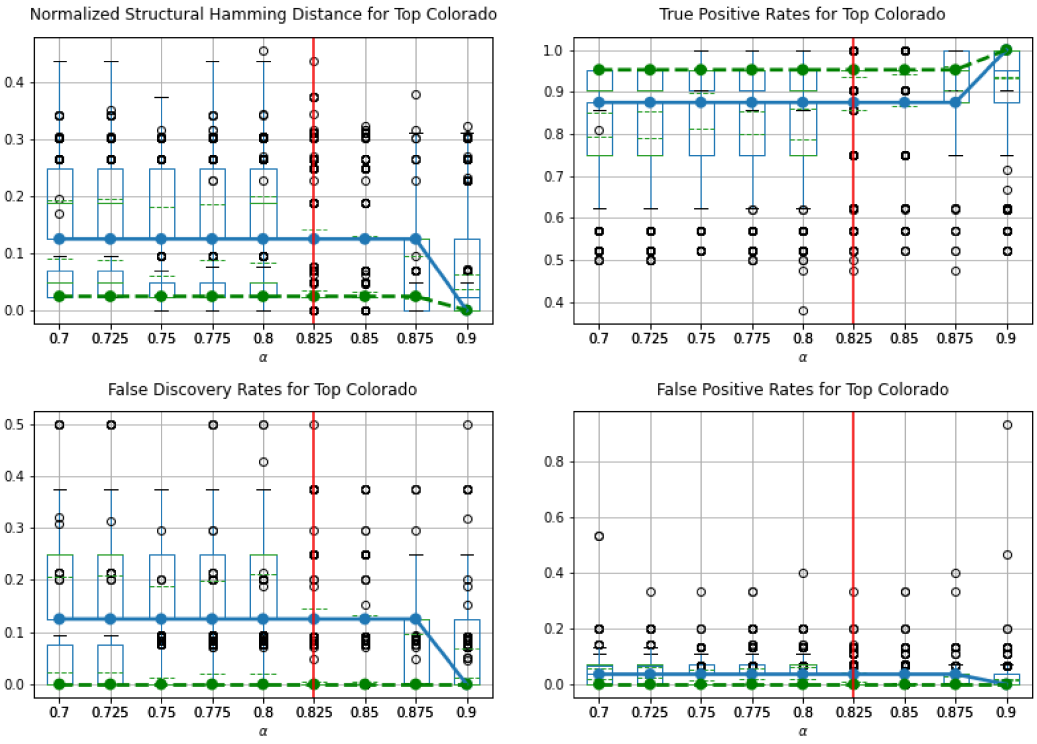

Fig. S4. Metrics nSHD, TPR, FDR and FPR for the Top sector of the Colorado network and varying parameters α. For detailed explanations see Figure 10 of the Paper.

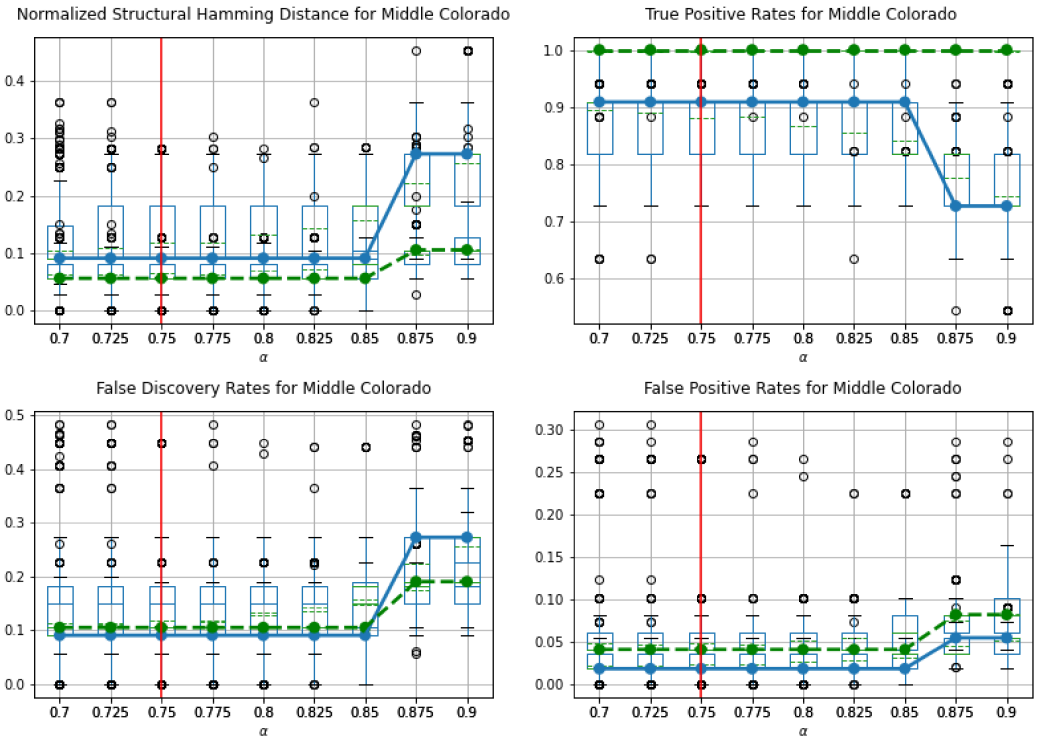

Fig. S5. Metrics nSHD, TPR, FDR and FPR for the Middle sector of the Colorado network and varying parameters α. For detailed explanations see Figure 10 of the Paper.

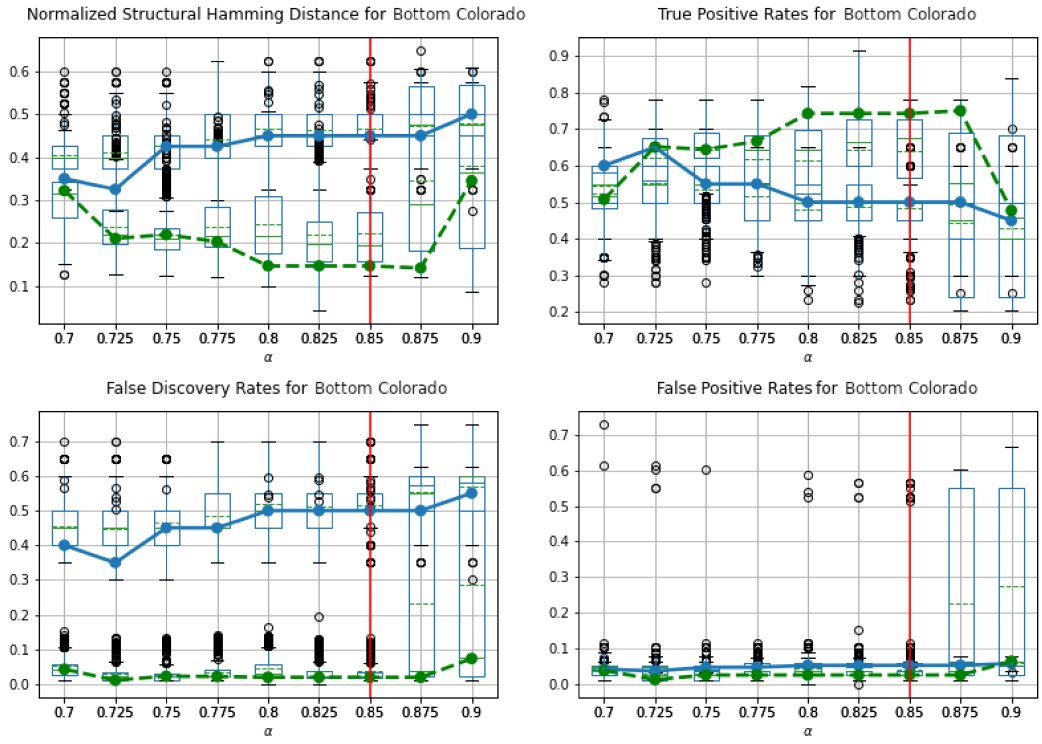

Fig. S6. Metrics nSHD, TPR, FDR and FPR for the Bottom sector of the Colorado network and varying parameters α. For detailed explanations see Figure 10 of the Paper.

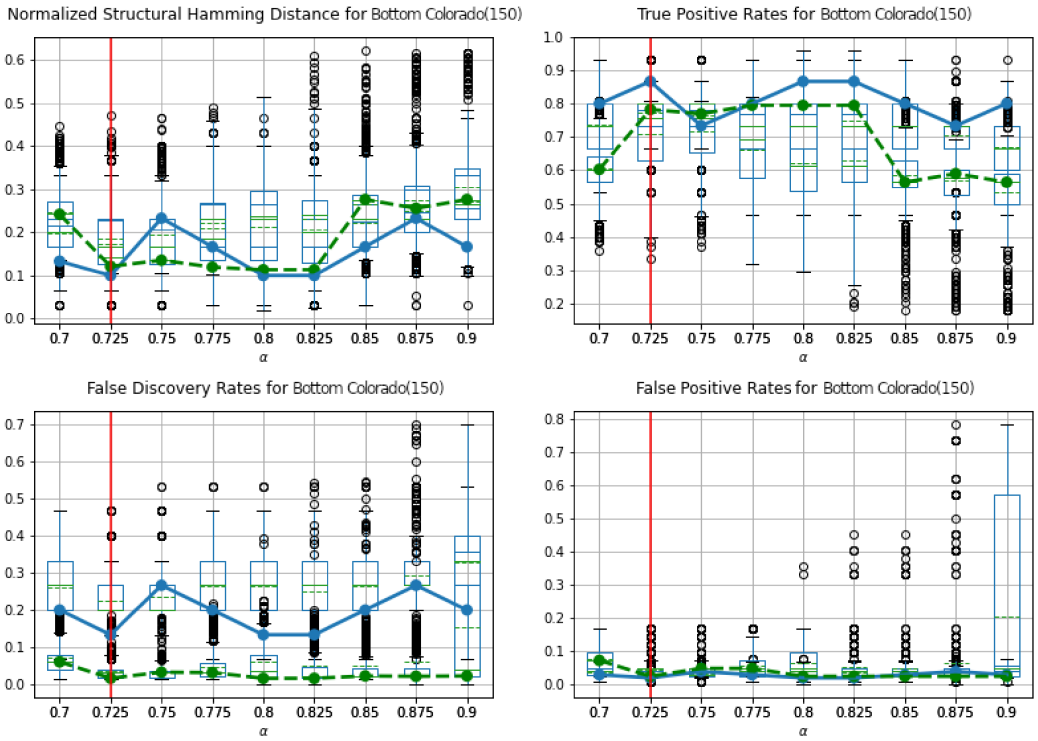

Fig. S7. Metrics nSHD, TPR, FDR and FPR for Bottom150 of the Colorado network and varying parameters α. For detailed explanations see Figure 10 of the Paper.

S4. Supplemental Figures for Section 5

Below we visualize the performance measures nSHD and TPR as defined in (1.1) of the Paper from simulations of settings (2) and (3) of Section 5 of the Paper.

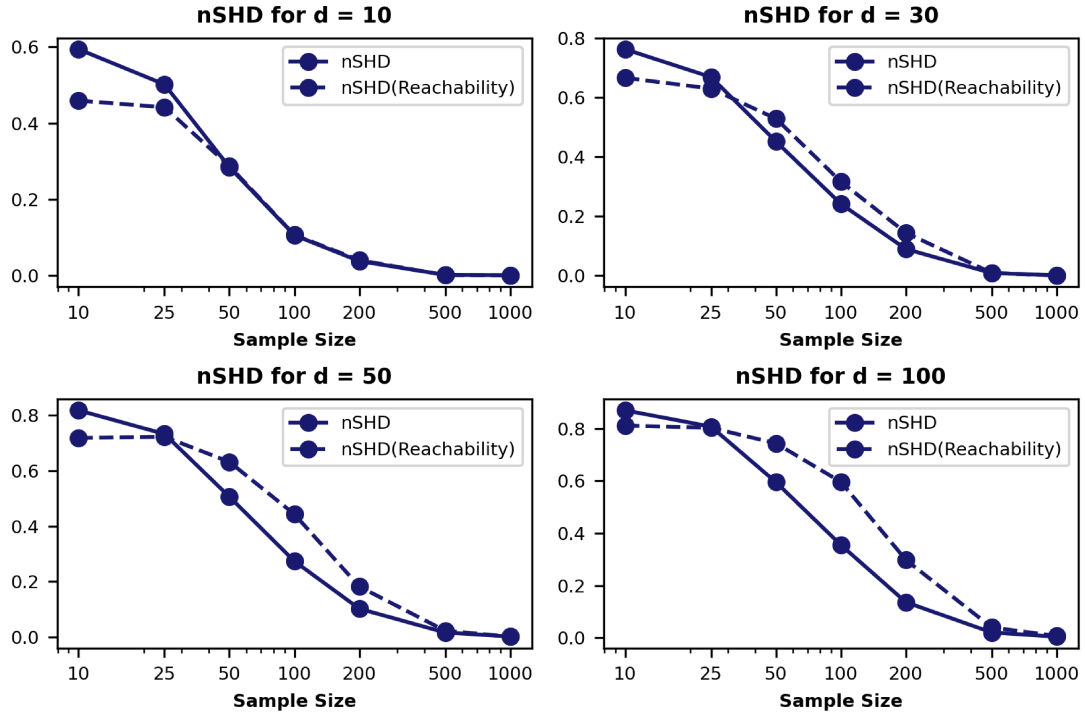

Fig. S8. Mean nSHD for the weak dependence setting (2) and different graph sizes. For detailed explanations, see Figure 12 of the Paper.

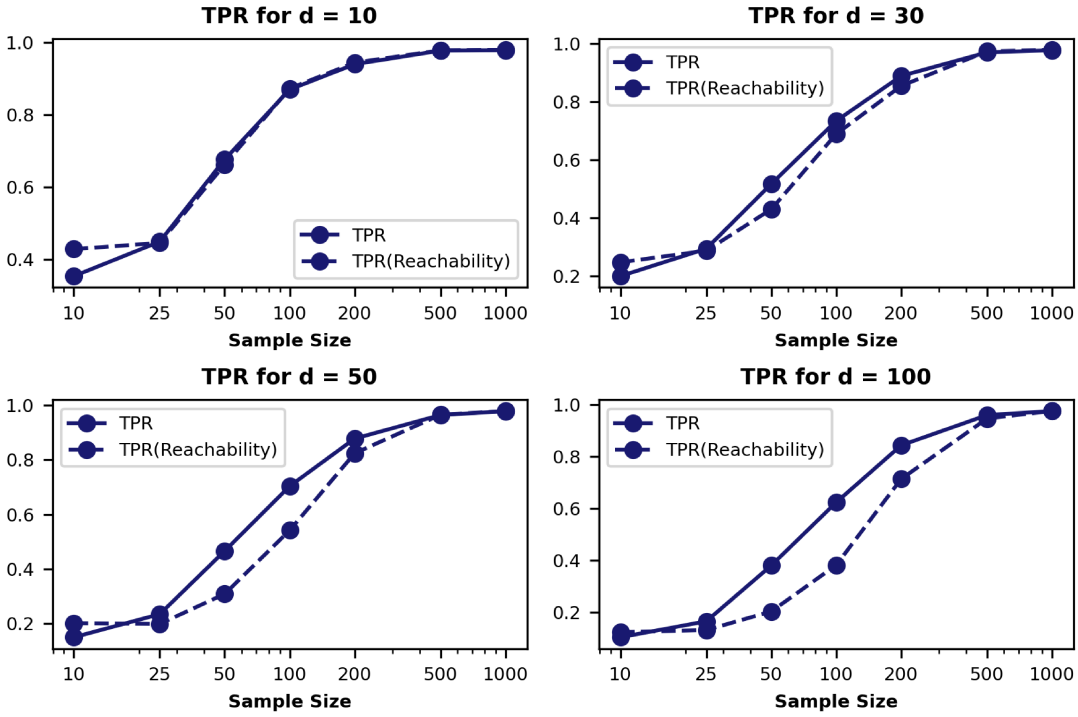

Fig. S9. Mean TPR for the weak dependence (2) and different graph sizes. For detailed explanations, see Figure 12 of the Paper.

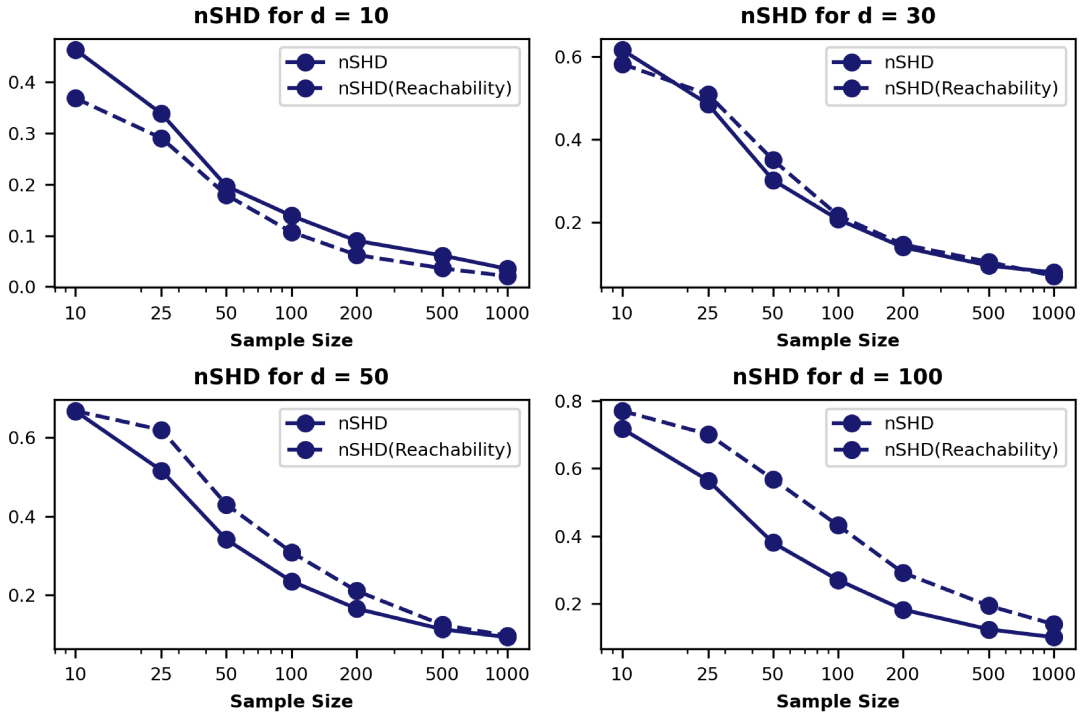

Fig. S10. Mean nSHD for the mixed distribution setting (3) and different graph sizes. For detailed explanations, see Figure 12 of the Paper.

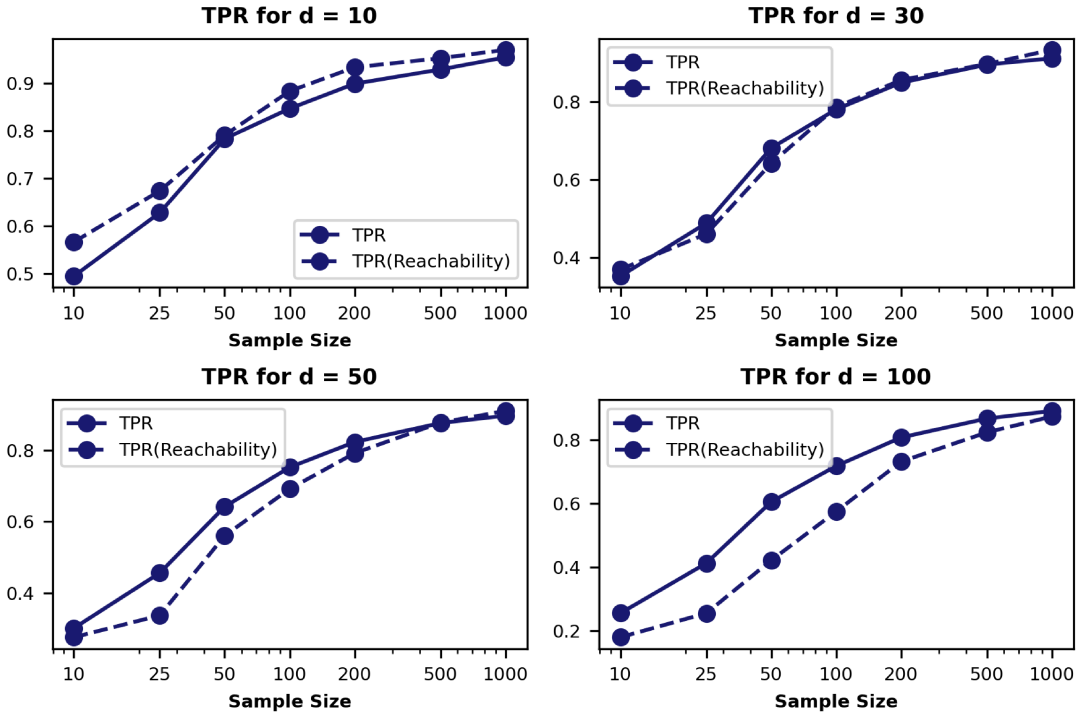

Fig. S11. Mean TPR for the mixed distribution setting (3) and different graph sizes. For detailed explanations, see Figure 12 of the Paper.

References

Améndola, C., Klüppelberg, C., Lauritzen, S. and Tran, N. M. (2022) Conditional independence in max-linear bayesian networks. *The Annals of Applied Probability*, **32**, 1–45.

Anderson, M. P., Woessner, W. W. and Hunt, R. J. (2015) *Applied Groundwater Modeling: Simulation of Flow and Advective Transport*. Academic Press.

Asadi, P., Davison, A. C. and Engelke, S. (2015) Extremes on river networks. *The Annals of Applied Statistics*, **9**, 2023–2050.

Asenova, S., Mazo, G. and Segers, J. (2021) Inference on extremal dependence in the domain of attraction of a structured hüsler-reiss distribution motivated by a markov tree with latent variables. ArXiv:2001.09510.

Asenova, S. and Segers, J. (2022) Max-linear graphical models with heavy-tailed factors on trees of transitive tournaments. ArXiv:2209.14938.

Baccelli, F., Cohen, G., Olsder, G. J. and Quadrat, J.-P. (1992) *Synchronization and Linearity: An Algebra for Discrete Event Systems*. John Wiley & Sons Ltd.

Balkema, A., Klüppelberg, C. and Resnick, S. (1993) Densities with Gaussian tails. *Proceedings of the London Mathematical Society*, **66**, 568–588.

Bartos, M., Wong, B. and Kerkez, B. (2018) Open storm: a complete framework for sensing and control of urban watersheds. *Environmental Science: Water Research & Technology*, **4**, 346–358.

Beirlant, J., Goegebeur, Y., Segers, J. and Teugels, J. (2004) *Statistics of Extremes: Theory and Applications*. Chichester: Wiley.

Bollen, K. A. (1989) *Structural Equations with Latent Variables*. New York: Wiley.

Boneh, A. and Hofri, M. (1997) The coupon-collector problem revisited—a survey of engineering problems and computational methods. *Stochastic Models*, **13**, 39–66.

Boneh, S. and Papanicolaou, V. G. (1996) General asymptotic estimates for the coupon collector problem. *Journal of Computational and Applied Mathematics*, **67**, 277–289.

Buck, J. and Klüppelberg, C. (2021) Recursive max-linear models with propagating noise. *Electronic Journal of Statistics*, **15**.

Chickering, D. M. (1996) Learning Bayesian networks is np-complete. In *Learning from data*, 121–130. Springer.

Coles, S., Bawa, J., Trenner, L. and Dorazio, P. (2001) *An Introduction to Statistical Modeling of Extreme Values*. Springer.

Coles, S., Heffernan, J. and Tawn, J. (1999) Dependence measures for extreme value analyses. *Extremes*, **2**, 339–365.

Davison, A. C. and Huser, R. (2015) Statistics of extremes. *Annual Review of Statistics and its Application*, **2**, 203–235.

Drton, M. T. and Maathuis, M. H. (2017) Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, **4**, 365–393.

Einmahl, J., Kiriliouk, A. and Segers, J. (2018) A continuous updating weighted least squares estimator of tail dependence in high dimensions. *Extremes*, **21**, 205–233.

Embrechts, P., Klüppelberg, C. and Mikosch, T. (1997) *Modelling Extremal Events for Insurance and Finance*. Berlin: Springer.

Engelke, S. and Hitz, A. (2020) Graphical models for extremes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **82**, 871–932.

Engelke, S., Hitz, A. S. and Gnecco, N. (2019) `graphicalExtremes`: Statistical Methodology for Graphical Extreme Value Models. URL: `https://CRAN.R-project.org/package=graphicalExtremes`. R package version 0.1.0.

Engelke, S., Lalancette, M. and Volgushev, S. (2022) Learning extremal graphical structures in high dimensions. ArXiv:2111.00840.

Engelke, S. and Volgushev, S. (2020) Structure learning for extremal tree models. ArXiv:2012.06179.

Gabow, H. N., Galil, Z., Spencer, T. and Tarjan, R. E. (1986) Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*, **6**, 109–122.

Gissibl, N. (2018) *Graphical Modeling of Extremes: Max-linear models on directed acyclic graphs*. Ph.D. thesis, Technical University of Munich.

Gissibl, N. and Klüppelberg, C. (2018) Max-linear models on directed acyclic graphs. *Bernoulli*, **24**, 2693–2720.

Gissibl, N. and Klüppelberg, C. (2018) Max-linear models on directed acyclic graphs. *Bernoulli*, **24**, 2693–2720.

Gissibl, N., Klüppelberg, C. and Lauritzen, S. L. (2021) Identifiability and estimation of recursive max-linear models. *Scandinavian Journal of Statistics*, **48**, 188–211.

Gissibl, N., Klüppelberg, C. and Otto, M. (2018) Tail dependence of recursive max-linear models with regularly varying noise variables. *Econometrics and Statistics*, **6**, 149 – 167.

Gnecco, N., Meinshausen, N., Peters, J. and Engelke, S. (2021) Causal discovery in heavy-tailed models. *The Annals of Statistics*, **49**, 1755–1778.

Gong, Y., Zhong, P., Opitz, T. and Huser, R. (2022) Partial tail-correlation coefficient applied to extremal-network learning. ArXiv:2210.07351.

Grötschel, M., Lovász, L. and Schrijver, A. (1988) *Geometric Algorithms and Combinatorial Optimization*. Heidelberg: Springer.

de Haan, L. and Ferreira, A. (2007) *Extreme Value Theory: An Introduction*. Springer Science & Business Media.

Hagberg, A. A., Schult, D. A. and Swart, P. J. (2008) Exploring network structure, dynamics, and function using networkX. In *Proceedings of the 7th Python in Science Conference* (eds. G. Varoquaux, T. Vaught and J. Millman), 11 – 15. Pasadena, CA USA.

Hu, S., Peng, Z. and Segers, J. (2022) Modelling multivariate extreme value distriubtions via markov trees. ArXiv:2208.02627.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013) *An Introduction to Statistical Learning: with Applications in R*. Springer.

Klüppelberg, C. and Krali, M. (2021) Estimating an extreme Bayesian network via scalings. *Journal of Multivariate Analysis*, **181**, 104672.

Klüppelberg, C. and Lauritzen, S. (2020) Bayesian networks for max-linear models. In *Network Science - An Aerial View from Different Perspectives* (eds. F. Biagini, G. Kauermann and T. Meyer-Brandis). Springer.

Larsson, M. and Resnick, S. I. (2012) Extremal dependence measure and extremogram: the regularly varying case. *Extremes*, **15**, 231–256.

Lauritzen, S. L. (1996) *Graphical Models*. Clarendon Press.

Leigh, C., Alsibai, O., Hyndman, R. J., Kandanaarachchi, S., King, O. C., McGree, J. M., Catherine Neelamraju, J. S., Talagala, P. D., Turner, R. D., Mengersen, K. and Peterson, E. E. (2019) A framework for automated anomaly detection in high frequency water-quality data from in situ sensors. *Science of the Total Environment*, **664**, 885–898.

Lim, L.-H. (2015) Hodge Laplacians on graphs. *SIAM Review*, **62**, 685–715.

Lower Colorado River Authority (LCRA) (2020a) Highland Lakes and Dams. Available at `https://www.lcra.org/water/dams-and-lakes`. Accessed August 2020.

— (2020b) Private correspondence.

Maathuis, M., Drton, M., Lauritzen, S. and Wainwright, M. (eds.) (2019) *Handbook of Graphical Models*. Chapman & Hall/CRC.

Mao, F., Khamis, K., Krause, S., Clark, J. and Hannah, D. M. (2019) Low-cost environmental sensor networks: recent advances and future directions. *Frontiers in Earth Science*, **7**, 221.

McGrane, S. J. (2016) Impacts of urbanisation on hydrological and water quality dynamics, and urban water management: a review. *Hydrological Sciences Journal*, **61**, 2295–2311.

Mhalla, L., Chavez-Demoulin, V. and Dupuis, D. J. (2020) Causal mechanism of extreme river discharges in the upper danube basin network. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **69**, 741–764.

Musser, D. R. (1997) Introspective sorting and selection algorithms. *Software: Practice and Experience*, **27**, 983–993.

Pearl, J. (2009) *Causality: Models, Reasoning, and Inference*. Cambridge: Cambridge University Press, 2nd edn.

Pearl, J. et al. (2009) Causal inference in statistics: An overview. *Statistics surveys*, **3**, 96–146.

Politis, D., Romano, J. and Wolf, M. (1999) *Subsampling*. New York: Springer.

Prim, R. (1957) Shortest connection networks and some generalizations. *Bell System Technical Journal*, **35**, 1389–1401.

Resnick, S. I. (1987) *Extreme Values, Regular Variation, and Point Processes*. New York: Springer.

— (2007) *Heavy-Tail Phenomena: Probabilistic and Statistical Modeling*. New York: Springer.

Rochet, J.-C. and Tirole, J. (1996) Interbank lending and systemic risk. *Journal of Money, Credit and Banking*, **28**, 733–762.

Rodriguez-Perez, J., Leigh, C., Liquet, B., Kermorvant, C., Peterson, E., Sous, D. and Mengersen, K. (2020) Detecting technical anomalies in high-frequency water-quality data using artificial neural networks. *Environmental Science & Technology*, **54**, 13719–13730.

Rötter, F., Engelke, S. and Zwiernik, P. (2022) Total positivity in multivariate extremes. ArXiv:2112.14727.

Segers, J. (2020) One-versus multi-component regular variation and extremes of markov trees. *Advances in Applied Probability*, **52**. To appear.

Sibuya, M. (1960) Bivariate extreme statistics. *Ann. Inst. of Statist. Math. Tokyo*, **11**, 195–210.

Spirtes, P., Glymour, C. N., Scheines, R. and Heckerman, D. (2000) *Causation, prediction, and search*. MIT press.

Tran, N. (2022) The tropical geometry of causal inference for extremes. ArXiv:2207.10227.

Tran, N. M. (2021) `QTree`: Python Implementation. URL: `https://github.com/princengoc/qtree`. GitHub repository version 0.1.0, Phyton Release 3.11.0.

Ver Hoef, J. M. and Peterson, E. (2010) A moving average approach for spatial statistical models of stream networks. *Journal of the American Statistical Association*, **105**, 6–18.

Ver Hoef, J. M., Peterson, E. and Theobald, D. (2006) Spatial statistical models that use flow and stream distance. *Environmental and Ecological Statistics*, **13**, 449–464.

Wainwright, M. J. and Jordan, M. I. (2008) *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc.

Wolf, L., Zwiener, C. and Zemann, M. (2012) Tracking artificial sweeteners and pharmaceuticals introduced into urban groundwater by leaking sewer networks. *Science of the Total Environment*, **430**, 8–19.

Zheng, X., Aragam, B., Ravikumar, P. K. and Xing, E. P. (2018) Dags with no tears: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, vol. 31, 9472–9483. Curran Associates, Inc.