Estimating a Latent Tree for Extremes

Ngoc Mai Tran*
Department of Mathematics, University of Texas at Austin
and
Johannes Buck and Claudia Klüppelberg†
Department of Mathematics, Technical University of Munich

August 24, 2021

Abstract

The Latent River Problem has emerged as a flagship problem for causal discovery in extreme value statistics. This paper gives QTree, a simple and efficient algorithm to solve the Latent River Problem that outperforms existing methods. QTree returns a directed graph and achieves almost perfect recovery on the Upper Danube, the existing benchmark dataset, as well as on new data from the Lower Colorado River 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. In addition, under a Bayesian network model for extreme values with propagating noise, we show that the QTree estimator returns for $n \to \infty$ a.s. the correct tree.

Keywords: causal inference, max-linear, Bayesian networks, extreme values statistics, directed graphical models

*ntan@math.utexas.edu
†j.buck@tum.de, cklu@tum.de
1 Introduction

Causal inference from extremes aims to discover cause and effect relations between large observed values of random variables. Such methods are much needed, for rare events are often interconnected. Credit markets might fail due to some endogenous systemic risk propagation [37]. Pollutants can propagate through an unseen underground waterway, causing extreme measurements at multiple locations. However, it is not immediately obvious how to extend the past decades of work on causal inference [8, 25, 35, 39] 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 [7, 14, 36], which are often difficult to estimate from limited data. For multivariate models, the combinatorial explosion in the joint likelihood makes them intractable to fit on large data sets even in moderate dimensions.

Over the last three years, almost all new methods for causal inference from extremes have been validated on the Latent River Problem [3, 16, 17, 22, 33]. 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 or terrain. Here, the true river network serves as the ‘gold standard’, allowing one to verify the performance of the proposed estimator. Success in solving the Latent River Problem can translate to new solutions to the contaminant tracing challenge in hydrology [26, 32, 38, 41, 42, 44]. 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 [2]. Recent advances point towards an imminent data explosion [6, 31], where pollutants exceeding certain thresholds can be detected via a sensor network. Thus, contaminant trac-
ing with sensors data is a version of the Latent River Problem without the gold standard, where the network is truly unknown.

The Latent River Problem has proven to be challenging and very stimulating for extreme value theory, with each paper taking a different technique: time-series [3], expected quantile scores [33], undirected graphical models for extremes [16], and causal tail coefficients [22]. On the benchmark Upper Danube data set [3], with $|V| = 31$ stations and 479 decorrelated river discharge maxima, [16] returned a highly accurate but undirected graph, [22] correctly recovered the main branch of the river (12 nodes out of 31) but did not learn the entire river network, while [33] did well at detecting stream-connected nodes, but often missed or wrongly estimated causal relations, see the table in Figure 6.

This paper presents QTree, a simple and efficient algorithm for solving the Latent River Problem under the assumption that the river network is a latent root-directed tree. It performs extremely well on real-world data sets. Fitted to 75% randomly chosen subsets of the data, QTree achieves almost perfect recovery on the Upper Danube, outperforming existing methods in the literature (cf. Figure 6). In addition to the Upper Danube, we further test QTree on three sections of the Lower Colorado in Texas [30]. These are much more challenging data sets. This river network suffers from severe drought, extreme flooding, and sensors failure, with up to 36.9% missing data (the Danube has none). These challenges make recovering the Lower Colorado much closer to the trace contaminant challenge. Remarkably, on all three sections of the Lower Colorado, QTree fitted to 75% subsamples also achieves almost perfect recovery (cf. Section 4).

Beyond hydrology, QTree can be applied to cause and effect detection in every high-risk problem such as financial risks (cf. [11, 15, 22, 23]), assuming that the latent 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 extremes \cite{1,19,20}, however, existing methods for learning them aim to learn the model parameters and thus are highly sensitive to model misspecifications \cite{19,21,24}. In particular, they have not performed 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 the Chu-Liu-Edmond algorithm to return the best root-directed spanning tree \cite{18}. We detail the algorithm and the intuition behind it in Section 2. Assuming that the data come from a noisy max-linear Bayesian network, we prove that the tree output by QTree is strongly consistent (cf. Theorem 2.1).

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. We also give QTree an automated parameter tuning procedure based on subsampling. We implemented QTree as a plug-and-play package in Python 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 QTree with automated parameter selection (Algorithm 2) in Section 2 and discuss its intuition supported by simulation results. In Section 3, we introduce the data sets, discuss their specific challenges and describe the data preprocessing steps. In Section 4, we give the estimation results of QTree and analyze the performance of the automated parameter selection. Section 5 concludes with a summary. The Supplementary Material includes the proof of the Consistency Theorem (cf. Theorem 2.1).
**Notations.** Estimators are compared based on standard metrics in causal inference\[15]: normalized structural Hamming distance (nSHD), false discovery rate (FDR), false positive rate (FPR), and true positive rate (TPR). All of these metrics lie between 0 and 1. Except for the TPR, the lower the number, the better the performance. We recall their definitions here. Let $G$ be the true graph, $\hat{\mathcal{G}}$ be the estimated graph. The *structural Hamming distance* $\text{SHD}(G, \hat{\mathcal{G}})$ between $G$ and $\hat{\mathcal{G}}$ is the minimum number of edge additions, deletions and reversals to obtain $G$ from $\hat{\mathcal{G}}$. Denote $E(G)$ and $E(\hat{\mathcal{G}})$ the set of edges in $G$ and $\hat{\mathcal{G}}$, respectively. Note that $|E(\hat{\mathcal{G}}) \setminus E(G)|$ is the number of edges in $\hat{\mathcal{G}}$ that are not in $G$, while $|E(\hat{\mathcal{G}}) \cap E(G)|$ is the number of correctly estimated edges. We then have

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

(1.1)

2 The algorithm

2.1 The data generation model

Let $\mathcal{T}$ be a root-directed spanning tree on $V$ nodes. That is, each node $i \in V$ except the root has *exactly one child*. In the Latent 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 \[20], a model for risk propagation over a directed acyclic graph. The model says that each edge $j \to i \in \mathcal{T}$ has a coefficient $c_{ij} > 0$, interpreted as some measure of flow rate from $j$ to $i$, and the extreme discharge at $i$ is either the result of an unknown external input $Z_i$ (e.g. heavy rainfall), or that it is the weighted maximum of
discharges from the parent nodes of \(i\) in \(\mathcal{T}\). That is,

\[
X_i = \bigvee_{j:j \to i \in \mathcal{T}} c_{ij}X_j \vee Z_i, \quad c_{ij}, Z_i \geq 0, i \in V.
\]  

(2.1)

For numerical stability, in practice one works with the logarithm of the extremes. To avoid new symbols, we keep the same notation, so the max-linear Bayesian network becomes

\[
X_i = \bigvee_{j:j \to i \in \mathcal{T}} (c_{ij} + X_j) \vee Z_i, \quad i \in V.
\]  

(2.2)

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

**The Latent River Problem.** Given \(n\) observations \(\mathcal{X} = \{x^1 + \varepsilon^1, \ldots, x^n + \varepsilon^n\}\) where the \(x\)'s are generated via (2.2), and the \(\varepsilon\)'s are independent noise in \(\mathbb{R}^d\), 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 [43]. 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 [12]. However, it follows from [20, Theorem 2.2] that 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 A.1 in the Supplementary Material). Here is the intuition. Fix an edge \(j \to i\). For
each vector of extreme discharges \( x \in \mathbb{R}^V \), either the extreme discharge at \( j \) causes that at \( i \), or it does not. In the noise-free model (2.2), if \( j \) does not cause \( i \), then \( x_i > c_{ij} + x_j \). If \( j \) causes \( i \), then \( x_i = c_{ij} + x_j \). Over \( n \) observations, if \( j \) causes \( i \) 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

\[
\mathcal{X}_{ij} := \{x_i - x_j : x \in \mathcal{X} \}
\]  

is concentrated at the minimum of its support if and only if \( j \rightsquigarrow i \), that is, \( j \) is an ancestor of \( i \). Thus, with enough observations, one can recover the ancestral edges \( j \rightsquigarrow i \), from which \( \mathcal{T} \) can be uniquely recovered as it is a root-directed tree.

The QTree algorithm 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 to see a concentration near the minimum of \( \mathcal{X}_{ij} \) iff \( j \rightsquigarrow i \). 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 (2.3) by the set

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

(2.4)

where \( Q_{\mathcal{X}_j}(\alpha) \) is the \( \alpha \)-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).

2.3 The QTree Algorithm

The QTree Algorithm (Algorithm 1) computes independently for each potential edge \( j \to i \) a score \( w_{ij} \), seen as a measure of dispersion of \( \mathcal{X}_{ij}(\alpha) \) near its minimum, then outputs the
For the simple DAG 1 → 2 with $c_{21} = \log(0.5) = -0.69$ and normal centered noise with standard deviation 0.5: First row: Histograms of the observations $X_{21}$ as in (2.3) (left) and truncated observations $X_{21}(0.8)$ as in (2.4). In most cases, for $X_{21}(0.8)$ 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 $X_{12}$ and $X_{12}(0.8)$ corresponding to the incorrectly directed edge. For $X_{12}(0.8)$ depicted in the right-hand figure, a large value $x_2$ is realised either from large $x_1 - x_2$ or from large $x_2$, giving the bimodal distribution.
minimum directed spanning tree of the graph $\mathcal{G}$ with weights $W = (w_{ij})$. The idea is that at each node $j$, the true edge would have the lowest dispersion (highest concentration) among all potential outgoing edges. Theorem 2.1 proves this for the Gumbel-Gaussian noise model. The default dispersion measure for $\text{QTree}$ is the quantile-to-mean gap

$$w_{ij}(r) := \frac{1}{n_{ij}} \left( \mathbb{E}(X_{ij}(\alpha)) - Q_{X_{ij}(\alpha)}(r) \right)^2,$$

(2.5)

where $r \in (0, 1)$ is a fixed, small quantile level and $n_{ij} = |\mathcal{X}_{ij}(\alpha)|$ is the number of observations in the set $X_{ij}(\alpha)$ defined in (2.4). The normalization factor $n_{ij}$ only matters when missing values are unevenly distributed across pairs, such as for the Lower Colorado network (cf. Section 3.1). Then, pairs with fewer observations get a relative penalty in the dispersion estimate to account for fluctuations in sample quantile estimates due to small observations. If no missing values are present, as is the case with the Danube, then $n_{ij} = n \cdot \alpha$ for all pairs $(i, j)$ and the algorithm would return the exact same tree $\hat{T}$ if dispersion was defined without dividing by $n_{ij}$.

**Algorithm 1 QTree for fixed parameters**

**Parameters:** $r \in (0, 1), \alpha \in [0, 1)$.

**Input:** data $\mathcal{X} = \{x^1, \ldots, x^n\} \subset \mathbb{R}^V$.

**Output:** a root-directed spanning tree $\hat{T}$ on $V$.

1: for $j \rightarrow i$, $j, i \in V, j \neq i$ do
2: Compute $w_{ij}(r)$ by (2.5).
3: Compute $\hat{T} :=$ minimum root-directed spanning tree on the directed graph $(V, \mathcal{G})$ with weight matrix $W = (w_{ij}(r)) \in \mathbb{R}^{V \times V}$ with the Chu-Liu-Edmond algorithm [13].
4: Return $\hat{T}$
We note that there are other choices for a dispersion measure, such as the lower quantile gap,
\[
w_{ij}(r, \overline{r}) := \frac{1}{n_{ij}} \left( Q_{X_{ij}(\alpha)}(r) - Q_{X_{ij}(\alpha)}(\overline{r}) \right)^2,
\]
where \(0 < r < \overline{r} < 1\) is a fixed pair of quantile levels. If \(r\) is small, then \(w_{ij}(r, \overline{r})\) is a local measure of dispersion in the lower tail of \(X_{ij}(\alpha)\). Note that, if the number of observations are small, then \(r\) cannot be too small, so the two empirical dispersion measures are in fact very 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.

### 2.3.1 Theoretical properties of QTee

Our Consistency Theorem, proved in the Supplementary Material, says that under the Gumbel-Gaussian noise model, both quantile-to-mean and lower quantile gap produce strongly consistent estimators for some 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 \(d = 100\), QTee only needs \(n = 200\) observations to bring the metrics TPR to more than 95% and nSHD as defined in (1.1) to less than 5%.

**Gumbel-Gaussian noise model.** For \(i \in V\), the sources \(Z_i\)'s in (2.2) are i.i.d. Gumbel(1) (location parameter 0 and scale parameter 1), the noises \(\varepsilon_i\)'s are i.i.d with a symmetric, light-tailed density \(f_\varepsilon\), satisfying some weak regularity condition on its derivative, given by
\[
f_\varepsilon(x) \sim e^{-Kx^p} \text{ as } x \to \infty,
\]
for some \(p > 1\) and \(\gamma, K > 0\).
Theorem 2.1 (Consistency Theorem). Assume the Gumbel-Gaussian noise model.

(a) There exists an \( r^* > 0 \) such that for any pair \( 0 < r < r^* \), the QTree algorithm with \( (w_{ij}) \) defined as the lower quantile gap \( w_{ij}(r, \tau) \) in (2.6) returns a strongly consistent estimator for the tree \( T \) as the sample size \( n \to \infty \).

(b) There exists an \( r^* > 0 \) such that for any \( 0 < r < r^* \), the QTree algorithm with \( (w_{ij}) \) defined as the quantile-to-mean gap \( w_{ij}(r) \) in (2.5) returns a strongly consistent estimator for the tree \( T \) as the sample size \( n \to \infty \).

Remark. The density (2.7) belongs to a special class of light-tailed densities whose convolution tail can be derived asymptotically [5]. The family includes the Gaussian \((p = 2)\), and though it is strictly more general than the Gaussian, we call our noise model Gumbel-Gaussian for ease of reference. Condition (2.7) guarantees that the upper tail of \( \varepsilon_i - \varepsilon_j \) is lighter than that of \( Z_i - Z_j \) (cf. Lemma B.4 in the Supplementary Material). To see why this condition is necessary, suppose we have just two nodes and suppose that the true graph is \( 1 \to 2 \). Then \( F_{21} \) is the distribution function of \( (\varepsilon_2 - \varepsilon_1) + (Z_2 - Z_1) \lor c_{21} \), while \( F_{12} \) is the distribution function of its negative. The lower tail of \( F_{21} \) essentially is the lower tail of \( \eta := (\varepsilon_2 - \varepsilon_1) \), while the upper tail is essentially the upper tail of the convolution \( (\varepsilon_2 - \varepsilon_1) + (Z_2 - Z_1) \), which would be 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 \( \eta \) has symmetric distribution, \( w_{12} = w_{21} \) in the second case, and it would be impossible to distinguish the edge \( 1 \to 2 \) from the edge \( 2 \to 1 \). In the first case, the asymmetry between the lower and upper tails of \( F_{12} \) lends us the crucial inequality that allows distinction between the two as illustrated in Figure I.
Figure 2: $1/(\text{mean errors})$ vs number of observations $n$ for different graph sizes $d = 30, 50, 100$. For each $d$ and each $n$, we simulated 100 data sets. To generate each data set, we generated a matrix $C$ by taking a uniform random spanning tree on $d$ nodes, choose a random node as the root, and set the edge weights to be i.i.d uniform on $[\log(0.2), \log(1)]$. Then we applied QTree with quantile-to-mean gap with $r = 0.05$ to estimate the true tree, and computed the average error as measured by 1-TPR (left) and nSHD (right). For a given $C$, noisy data $\{x_1, \ldots, x_n\}$ was simulated under the Gumbel-Gaussian model. At each node $i \in V$, the Gaussian noise has variance equal to 0.3 of the variance of the noise-free part, giving a 30% noise-to-signal ratio.

2.4 Parameter tuning using cross-validation

2.4.1 The trade-off intuition

Algorithm 1 has two parameters: the quantile $r \in (0, 1)$ and the cut-off quantile $\alpha \in (0, 1)$. If the data perfectly came from a max-linear Bayesian tree, then we should select $r$ as small as possible and base QTree on all data $\mathcal{X}$. However, due to the presence of noise, setting $r$ too small would make the estimator volatile to large values of the noise variables. Moreover, if the model was correctly specified, the larger $\alpha$ is, the higher the data concentration of $\mathcal{X}_{ij}(\alpha)$ would be at its minimum. On the other hand, the larger $\alpha$ is chosen, the influence of the light-tailed noise variables is dominated by the Gumbel(1) innovations, such that the
upper order statistics of the sample are close to the upper order statistics of a Gumbel(1) distribution, so to the correctly specified model. However, also fewer observations are available to estimate its dispersion near a low quantile.

2.4.2 Auto-tuned QTree

Here we propose a cross-validation procedure to automatically choose $r$ and $\alpha$ in QTree. This results in Algorithm 2, which we refer to as auto-tuned QTree.

Our key indicator for model performance is variability in the estimated tree, that is, whether the tree $\hat{T}$ and its reachability graph $\hat{R}$ output by QTree would change significantly if we fit it to different subsamples of the data. We propose the following definition of variability for a distribution of directed spanning trees.

**Definition 2.2.** Let $V$ be a set of nodes and $T = \{T^1, \ldots, T^m\}$ a collection of directed spanning trees on $V$, and let $R = \{R^1, \ldots, R^m\}$ be their corresponding reachability graphs. The centroid of $T$, denoted $E(T)$, is the root-directed spanning tree on $V$ that minimizes the normalized structural Hamming distance $d_H$ from $[1,1]$ to a randomly chosen tree; that is

$$E(T) := \arg \min \left\{ \sum_{i=1}^{m} d_H(T, T^i) : T \text{ a randomly chosen directed spanning tree on } V \right\}$$

(2.8)

Let $E(R)$ denote the reachability graph of $E(T)$. Let $e_T$ be the number of edges of $E(T)$, and $e_R$ be the number of edges of $E(R)$, respectively. The variability of $T$, denoted $\text{Var}(T)$, is

$$\text{Var}(T) := \frac{1}{e_T} \frac{1}{m} \sum_{i=1}^{m} d_H(T^i, E(T)) + \frac{1}{e_R} \frac{1}{m} \sum_{i=1}^{m} d_H(R^i, E(R)).$$

(2.9)
The reachability graph in (2.9) penalizes the situation where $T^i$ and $E(T)$ 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 $E(T)$ is the maximum directed spanning tree of a particular graph with weight matrix $S(T)$ that measures the stability among the trees in $T$. In particular, $E(T)$ can be computed using the Chu-Liu-Edmond algorithm [13], and thus $\text{Var}(T)$ can be computed in polynomial time.

**Lemma 2.3.** Let $V$ be a set of nodes and $T = \{T^1, \ldots, T^m\}$ be a collection of directed spanning trees on $V$. Define the stability weight matrix $S := S(T) \in \mathbb{R}_{\geq 0}^{d \times d}$ by

$$s_{ij} := S(T)_{ij} := \# \{T \in T : j \rightarrow i \in T \}. \quad (2.10)$$

Suppose that the maximum directed spanning tree $T_{\text{max}}$ of the graph on $V$ with weight matrix $S(T)$ is unique. Then $E(T) = T_{\text{max}}$.

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

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

which would establish that $T_{\text{max}} = E(T)$ by (2.8). Indeed,

$$\sum_{i=1}^{m} d_H(T', T^i) = \sum_{i=1}^{m} \sum_{u,v \in V : u \neq v} 1 \{T'_{uv} \neq T^i_{uv}\} = \sum_{u,v \in V : u \neq v} \sum_{i=1}^{m} 1 \{T'_{uv} \neq T^i_{uv}\}$$

$$= \sum_{u,v \in V : u \neq v} (s_{uv} 1_{u \rightarrow v \notin T'} + (m - s_{uv}) 1_{u \rightarrow v \in T'})$$

14
\[ \langle S, 1 - T' \rangle + \langle m1 - S, T' \rangle = -2\langle S, T' \rangle + \langle S, 1 \rangle + \langle m1, T' \rangle \]
\[ = -2\langle S, T' \rangle + \langle S, 1 \rangle + m(d - 1) \quad \text{since } T' \text{ is a spanning tree on } V \]
\[ \geq -2\langle S, T_{\max} \rangle + \langle S, 1 \rangle + m(d - 1) \quad \text{by definition of } E(T) \]
\[ = -2\langle S, T_{\max} \rangle + \langle S, 1 \rangle + \langle m1, T_{\max} \rangle \quad \text{since } T_{\max} \text{ is a spanning tree on } V \]
\[ = \sum_{i=1}^{m} d_H(T_{\max}, T^i). \]

As shown in Lemma A.2 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 dispersion measures in (2.5) or (2.6) 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 2.4.** Algorithm 2 has complexity \(O(|V|^2 nm|\Theta|)\).

**Proof.** For each pair \((r, \alpha) \in \Theta\), step 3 takes \(O(mn)\) and step 4 takes \(O(|V|^2 nm)\) by Lemma A.2 in the Supplementary Material. Step 6 takes \(O(m|V|^2)\), and step 7 takes \(O(|V|^2)\) by Chu-Liu-Edmond. Computing the reachability graph for a 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 \(E(T)\). So, for each pair \((r, \alpha) \in \Theta\), steps 3 to 8 take \(O(|V|^2 nm)\) time. Thus overall, the algorithm complexity is \(O(|V|^2 nm|\Theta|)\). \(\square\)
Algorithm 2 QTree with automated parameter selection

**Parameters**: subsampling fraction $f \in [0, 1]$, number of repetitions $m \in \mathbb{N}$, a set of parameters $\Theta = \{(r, \alpha)\} \subset [0, 1)^2$ to search over.

**Input**: data $\mathcal{X} = \{x^1, \ldots, x^n\} \subset \mathbb{R}^V$.

**Output**: the optimal parameter $(r^*, \alpha^*) \in \Theta$ and the corresponding root-directed spanning tree $\hat{T}_{\text{max}}$ on $V$.

1: for $(r, \alpha) \in \Theta$ do
2:   for $\ell = 1, \ldots, m$ do
3:       Sample without replacement a random subset $\mathcal{X}^\ell$ of $n \cdot f$ observations from $\mathcal{X}$
4:       Let $T^\ell$ be the output of QTree($r, \alpha$) fitted on $\mathcal{X}^\ell$.
5:   Let $T(r, \alpha) = \{T^\ell : \ell = 1, \ldots, m\}$
6:   Compute $S(T(r, \alpha))$ by (2.10)
7:   Compute $E(T(r, \alpha))$ as the maximum directed spanning tree of $S(T(r, \alpha))$ per Lemma 2.3.
8:   Compute $\text{Var}(T(r, \alpha))$ by (2.9)
9: Define $(r^*, \alpha^*) := \arg \min \{\text{Var}(T(r, \alpha)) : (r, \alpha) \in \Theta\}$.
10: **Return** the optimal pair $(r^*, \alpha^*)$ and $\hat{T}_{\text{max}} := E(T(r^*, \alpha^*))$. 

16
3 Data description

In general, some preprocessing is needed to turn raw daily extreme discharges into uncorrelated extreme discharge events on a river. This was detailed in Asadi et al [3] for the Upper Danube data set. The idea is to find non-overlapping time windows of \( p \) days, centered around the observation of maximal rank across all series. For each series, one then takes the maximum within the given time window, deletes the data of this window, and proceeds until no window of \( p \) consecutive days remains. We repeat that protocol for all three sections of the Lower Colorado in Texas to produce new data sets for the Latent River Problem. The Upper Danube data set as well as the data sets of all three sections of the Lower Colorado in Texas are available at https://github.com/princengoc/qtree. This section details how we carried out the preprocessing from the raw data provided by the Lower Colorado River Authority (LCRA, https://www.lcra.org/). For the Upper Danube, we use the preprocessed data provided by Asadi et al [3].

3.1 The Lower Colorado in Texas

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 particular interest. We use daily river discharge data from the 1st of December 1991 to the 14th of April 2020 (10,363 days) measured in cubic feet per second (cfs). A particularly challenging feature of the Lower Colorado is prolonged drought (discharge at 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 sets much more challenging than the Upper Danube data set.
In the Lower Colorado Basin, multiple dams cut off the river into disjoint sections \[29\], and many nodes have excessive amounts of missing data. Thus, we split the river 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 stream-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 maximum 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 gives three data sets, which we call the \textit{Top, Middle and Bottom} sectors of the Lower Colorado, with 9, 12 and 21 nodes, respectively (cf. Figure 5). From here on we treat these three sections as three separated, unrelated data sets.
Figure 4: A typical discharge at various nodes around one storm event on the Lower Colorado. The vertical lines mark a time window of $p = 8.5$ days. In nodes 3, 4 and 6, the thick dot denotes the peak discharge during this window. In node 5, the sensor did not function during this entire time period, so the peak for node 5 is recorded as missing. Over the entire study period (1991 to 2020, 10,363 days), node 3 has a median discharge of only 15 cfs and is hence mostly drained, but the water flow regularly aggregates to over 16,000 cfs within a very short period of time.

For each data set, we apply the declustering procedure of Asadi et al [3]. However, to account for sensor failures, we take maxima of each 12-hour period, then choose the declustering window $p = 8.5$ days. 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 storm is over [28], and thus the sensor potentially did not measure the largest possible water discharge that occurred at node $i$. 
Figure 5: The Top (upper left), Bottom (upper right) and Middle (lower) sections of the Lower Colorado network, treated as three unrelated data sets. The Top section has 9 nodes, 977 observations, 15.6% missing overall. The same statistics for the Middle section are 12 nodes, 973 observations, 26% missing, respectively. The Bottom section is most challenging, for it has the most nodes (21 nodes), the highest amount of missing data (35.5 %), and many nodes around the city of Austin with only a few miles apart from each other. The close proximity of these nodes induce strong spatial dependence even among nodes that are not stream-connected, making it potentially more challenging to recover the true network.
4 Results

For each of the four river networks (Danube, Top, Middle and Bottom section of the Colorado), we ran auto-tune QTree (Algorithm 2) with fixed $r = 0.05$, subsampling rate $f = 0.75$, and number of repetitions $m = 1000$ to choose $\alpha$ automatically from \{0.7, 0.725, 0.75, ..., 0.9\}. The main reason we fixed $r$ and varied $\alpha$ is for ease of visualization, as the performance of the automatic tuning procedure and sensitivity of QTree to $\alpha$ can be visualized with a two-dimensional plot (cf. Figures 10 and 11). The optimal parameters $\alpha^*$ selected by QTree for these networks are shown in Table 1.

|          | Danube | Bottom | Middle | Top  |
|----------|--------|--------|--------|------|
| $\alpha^*$ | 0.775  | 0.85   | 0.75   | 0.825|

Table 1: Optimal parameters $\alpha^*$ selected by QTree using grid search with cross-validation.

We find that in all data sets, QTree is not too sensitive to the choice of $\alpha$: there is often a range of $\alpha$ that result in the same estimated tree (cf. Figures 10 and 11 and Figures 16 to 18 in the Supplementary Material). The $\alpha^*$ chosen by the automated tuning procedure results in a tree that is either optimal (Middle and Bottom sections) or differs from the optimal tree by just one or two edges (Top section and the Danube).

Figures 6 to 8 show the estimated trees of the Danube, Top, Middle and Bottom sections 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 (1.1).
As can be seen visually, 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, FDR and FPR are below 9% and TPR is over 87%. In other words, a false edge tends to be from an ancestor to a child, rather than a spurious edge. QTree estimate of the Danube is substantially better than the state-of-the-art [33] across all metrics.

Figure 8 visualizes the estimation of the Bottom section 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 wrong edges, between 6537 and 24 and between 42 and 5525. Both of these node pairs are physically close. All the remaining incorrect edges are stream-connected. We note that the majority of errors made by QTree involve the 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 $\alpha^*$ 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. The result (cf. Figure 8) shows significant improvements. This shows 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.

As expected from a statistical estimation procedure, the statistical choice of the parameter selection by auto-tuned QTree sometimes does not output the best result, though it gets to within one or two edges of the optimal on every data set.
Figure 6: The Danube network, estimated by QTree vs true. Solid edges are correct. Dashed edges are not in the tree but are in the reachability graph, that is, the causal direction or stream-connection is correct. Squiggly edges are wrong (neither in the tree nor the reachability graph). Dotted edges are present in the true network but not in the estimated. The QTree outputs a tree with only six incorrect edges, four of them stream-connected and skipping only a single node (the edge 8 → 6 skips node 7). In the performance table, nSHD, TPR and FDR are defined as in equation (1.1). Numbers display the respective metric for the pair \((T, \hat{T})\) and numbers in brackets for the pair \((R, \hat{R})\) of their respective reachability graphs. Observe that this is a significant improvement over the state-of-the-art results of [33] based on expected quantile scores (EQS) (see [33, Figure 7]) as also seen in the performance metrics. We note that [3] achieves similar accuracy but only for the less challenging problem of estimating an undirected graph.

| e = j \(\rightarrow\) i : e ∈ \(\hat{T} \cap T\) | nSHD | 0.18(0.09) | 0.77(0.56) |
| e = j \(\rightarrow\) i : e ∈ \(\hat{T} \cap R \setminus T\) | TPR | 0.80(0.87) | 0.26(0.29) |
| e = j \(\rightarrow\) i : e ∈ \(\hat{T} \setminus R\) | FDR | 0.20(0.05) | 0.80(0.10) |
| e = j \(\rightarrow\) i : e ∈ \(\hat{T} \setminus T\) | |

QTree | EQS
Figure 7: Estimated Top (left) and Middle sections of the Colorado (right). Node colors represent the amount of available data after declustering. Arrows and performance table are as described in Figure 6. Both estimated networks only contain one single incorrect edge.
Figure 8: Estimated Bottom section of the Colorado (untrimmed and trimmed). Top Figure: untrimmed, based on all 21 nodes, QTree outputs a tree with ten incorrect edges, eight of them stream-connected. Bottom Figure: trimmed, based on 16 nodes, after removing nodes with less than 150 observations. We refer to it as Bottom150. There are only two incorrect edges, one of which stream-connected. Compared to the full Bottom section of the Colorado, this is a significant improvement. Node colors represent the amount of available data after declustering. Arrows and performance table are as described in Figure 6.
Figure 9: Estimated Danube network for $\alpha = 0.75$ (compare to Figure 6). Node colors represent the amount of available data after declustering. Arrows and performance table are as described in Figure 6.

For example, for the Danube network the parameter $\alpha = 0.75$ (instead of the optimal $\alpha^* = 0.775$) would have lead to a much better result (cf. Figure 9). Also for the Top Colorado, $\alpha = 0.9$ would have given perfect recovery of the underlying network, see Figure 16 of the Supplementary Material.

In summary, on all four data sets considered, QTree performed well for nodes with a sufficient number of observations. The chosen tree is either optimal (Middle and Bottom sections), or within one to two edges of the optimal (Danube and Top section). The method can handle data with a large fraction of missing observations and close spatial proximity between nodes.
Figure 10: Metrics nSHD, TPR, FDR and FPR for QTree outputs \( \hat{T}_\alpha \) for varying parameters \( \alpha \) for the Danube network. The solid lines show the interpolated metrics for \((\hat{T}_\alpha, T)\) and the dashed lines for \((\hat{R}_\alpha, R)\). We sampled without replacement 75\% of the data and repeated this 1000 times, creating 1000 data sets. For each \( \alpha \), we fitted QTree on these data sets to obtain 1000 estimated trees \( T_\alpha := \{\hat{T}_\alpha^1, \ldots, \hat{T}_\alpha^{1000}\} \). We compute the respective metric for each tree in \( T_\alpha \) (presented as boxplots). The tree returned by QTree for this \( \alpha \) is the centroid \( E(T_\alpha) \) (cf. (2.8)). By definition, the chosen \( \alpha^* \) is the parameter with the least variability in \( T_\alpha \) (vertical line).
Figure 11: Metrics nSHD, TPR, FDR and FPR for the Bottom section of the Colorado for varying parameters $\alpha$. For detailed explanations see Figure 10.
5 Summary

In this paper, we gave auto-tuned QTree, an algorithmic solution to the Latent River Problem, a benchmark problem for causal inference for extremes. We presented three new datasets for the Latent River Problem, which are more challenging than the classic Danube data due to a large fraction of missing data and close spatial proximity between nodes. Across all four datasets, auto-tuned QTree performed very well, significantly better than previous state-of-the-art results. The method can handle data with a large fraction of missing observations. Our plug-and-play Python implementation at [40] can fit QTree on ten thousand samples in the range of 10 to 30 nodes on a personal laptop within half an hour. We proved that under the max-linear Bayesian network with Gumbel-Gaussian distributions for sources and noise, QTree is consistent. We expect that QTree will become the algorithmic benchmark for causal inference for extremes.

Open research directions include (i) generalizations to learning latent directed acyclic graphs, (ii) better cross-validation metrics with theoretical guarantees, and (iii) have the algorithm output a distribution over possible root-directed trees instead of a single best tree.

Acknowledgements. We thank the Lower Colorado River Authority for providing the original data. We are also grateful to the authors of [3] for providing Figure 3 of the Danube Basin as well as the declustered data. Johannes Buck is financially supported by the Hanns-Seidel Foundation. Ngoc Tran is supported by NSF Grant DMS-2113468 and the NSF IFML 2019844 award. She also gratefully acknowledges support by the Hausdorff Center for Mathematics and the University of Bonn, Germany.
References

[1] C. Améndola, C. Klüppelberg, S. Lauritzen, and N. M. Tran. Conditional independence in max-linear Bayesian networks. *Ann. Appl. Prob.*, 2020. to appear. arXiv:2002.09233v2.

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

[3] P. Asadi, A. C. Davison, and S. Engelke. Extremes on river networks. *Annals of Applied Statistics*, 9(4):2023–2050, 2015.

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

[5] A. Balkema, C. Klüppelberg, and S. Resnick. Densities with Gaussian tails. *Proceedings of the London Mathematical Society*, 66(3):568–588, 1993.

[6] M. Bartos, B. Wong, and B. Kerkez. Open storm: a complete framework for sensing and control of urban watersheds. *Environmental Science: Water Research & Technology*, 4(3):346–358, 2018.

[7] J. Beirlant, Y. Goegebeur, J. Segers, and J. Teugels. *Statistics of Extremes: Theory and Applications*. Wiley, Chichester, 2004.

[8] K. A. Bollen. *Structural Equations with Latent Variables*. Wiley, New York, 1989.

[9] A. Boneh and M. Hofri. The coupon-collector problem revisited—a survey of engineering problems and computational methods. *Stochastic Models*, 13(1):39–66, 1997.
[10] S. Boneh and V. G. Papanicolaou. General asymptotic estimates for the coupon collector problem. *Journal of Computational and Applied Mathematics*, 67(2):277–289, 1996.

[11] J. Buck and C. Klüppelberg. Recursive max-linear models with propagating noise. Submitted. arXiv:2003.00362, 2020.

[12] D. M. Chickering. Learning Bayesian networks is np-complete. In *Learning from data*, pages 121–130. Springer, 1996.

[13] Y. Chu and T. Liu. On the shortest arborescence of a directed graph. *Science Sinica*, (14):1396–400, 1965.

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

[15] J. H. Einmahl, A. Kiriliouk, and J. Segers. A continuous updating weighted least squares estimator of tail dependence in high dimensions. *Extremes*, 21(2):205–233, 2018.

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

[17] S. Engelke and J. Ivanovs. Sparse structures for multivariate extremes. arXiv:2004.12182, 2020.

[18] H. N. Gabow, Z. Galil, T. Spencer, and R. E. Tarjan. Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*, 6(2):109–122, 1986.
[19] N. Gissibl. *Graphical Modeling of Extremes: Max-linear models on directed acyclic graphs*. Ph.D. thesis, Technical University of Munich, 2018.

[20] N. Gissibl and C. Klüppelberg. Max-linear models on directed acyclic graphs. *Bernoulli*, 24(4A):2693–2720, 2018.

[21] N. Gissibl, C. Klüppelberg, and S. L. Lauritzen. Identifiability and estimation of recursive max-linear models. *Scandinavian Journal of Statistics*, 48(1):188–211, 2021. doi: 10.1111/sjos.12446.

[22] N. Gnecco, N. Meinshausen, J. Peters, and S. Engelke. Causal discovery in heavy-tailed models. *Annals of Statistics*, to appear, 2021.

[23] C. Klüppelberg and M. Krali. Estimating an extreme Bayesian network via scalings. *Journal of Multivariate Analysis*, 181:104672, 2021. doi.org/10.1016/j.jmva.2020.104672.

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

[25] S. L. Lauritzen. *Graphical Models*. Clarendon Press, 1996.

[26] C. Leigh, O. Alsibai, R. J. Hyndman, S. Kandanaarachchi, O. C. King, J. M. McGree, J. S. Catherine Neelamraju, P. D. Talagala, R. D. Turner, K. Mengersen, and E. E. Peterson. A framework for automated anomaly detection in high frequency water-quality data from in situ sensors. *Science of the Total Environment*, 664(5):885–898, 2019.
[27] L.-H. Lim. Hodge Laplacians on graphs. *SIAM Review*, 62, 07 2015.

[28] Lower Colorado River Authority (LCRA). Private correspondence, 2020.

[29] Lower Colorado River Authority (LCRA). Highland lakes and dams, Accessed August 2020. [https://www.lcra.org/water/dams-and-lakes](https://www.lcra.org/water/dams-and-lakes).

[30] Lower Colorado River Authority (LCRA). Lower Colorado River Authority dataset, accessed August 2020. [https://hydromet.lcra.org](https://hydromet.lcra.org).

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

[32] S. J. McGrane. Impacts of urbanisation on hydrological and water quality dynamics, and urban water management: a review. *Hydrological Sciences Journal*, 61(13):2295–2311, 2016.

[33] L. Mhalla, V. Chavez-Demoulin, and D. J. Dupuis. Causal mechanism of extreme river discharges in the upper danube basin network. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 69(4):741–764, 2020.

[34] D. R. Musser. Introspective sorting and selection algorithms. *Software: Practice and Experience*, 27(8):983–993, 1997.

[35] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2nd edition, 2009.

[36] S. I. Resnick. *Heavy-Tail Phenomena: Probabilistic and Statistical Modeling*. Springer, New York, 2007.
[37] J.-C. Rochet and J. Tirole. Interbank lending and systemic risk. *Journal of Money, Credit and Banking*, 28(4):733–762, 1996.

[38] J. Rodriguez-Perez, C. Leigh, B. Liquet, C. Kermorvant, E. Peterson, D. Sous, and K. Mengersen. Detecting technical anomalies in high-frequency water-quality data using artificial neural networks. *Environmental Science & Technology*, 54(21):13719–13730, 2020.

[39] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman. *Causation, prediction, and search*. MIT press, 2000.

[40] N. M. Tran. QTree Python Implementation. [https://github.com/princengoc/qtree](https://github.com/princengoc/qtree), 2021.

[41] J. M. Ver Hoef and E. Peterson. A moving average approach for spatial statistical models of stream networks. *Journal of the American Statistical Association*, 105(489):6–18, 2010.

[42] J. M. Ver Hoef, E. Peterson, and D. Theobald. Spatial statistical models that use flow and stream distance. *Environmental and Ecological Statistics*, 13(4):449–464, 2006.

[43] M. J. Wainwright and M. I. Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc, 2008.

[44] L. Wolf, C. Zwiener, and M. Zemann. Tracking artificial sweeteners and pharmaceuticals introduced into urban groundwater by leaking sewer networks. *Science of the Total Environment*, 430:8–19, 2012. doi: [https://doi.org/10.1016/j.scitotenv.2012.04.059](https://doi.org/10.1016/j.scitotenv.2012.04.059).
A Proof of the complexity of QTrees

We work with the solution of the max-linear Bayesian network on a DAG $\mathcal{D}$ [4, §3]. Let $C^* = (c^*_{ij})$ be the matrix of longest paths, also known as the Kleene star of $C = (c_{ij})$. Then

$$X_i = \bigvee_{j \rightarrow i \in \mathcal{D}} (c^*_{ij} + Z_j), \quad i \in V. \quad \text{(A.1)}$$

Lemma A.1 concerns the noise-free case, and Lemma A.2 concerns the noisy case.

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

$$\hat{c}_{ij} = \min_{x \in \mathcal{X}} (x_i - x_j). \quad \text{(A.2)}$$

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

$$\hat{c}_{ij} = c^*_{ij} \iff \min \text{ in (A.2)} \text{ is achieved at least twice.} \quad \text{(A.3)}$$

In particular, $C^*$ can be computed in time $O(|V|^2 n)$. If one assumes that at each node, the parents are independently and equally likely to be the one that achieves the maximum, then for $n = O(|V|(|\log(|V|)|)^2)$ observations, one has a high probability of recovering $C^*$ exactly.

35
Proof. Note that $x_i \geq c^*_{ij} + x_j$ for all $x \in X$, and if $j \rightarrow i$ in observation $x$, then $x_i = c^*_{ij} + x_j$. Rearrange, we get (A.2). Equation (A.3) is [20, Theorem 2.2]. 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 (A.2), the complexity is $O(|V|^2n)$. The number of observations needed is so that each edge is seen at least twice is a variant of coupon-collecting [9, 10], 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 A.2 (Complexity of QTree). The QTree Algorithm runs in time $O(|V|^2n)$.

Proof. For each pair $i, j \in V, i \neq j$, to estimate $w_{ij}$, one needs to compute the $\alpha$-th quantile of $X_j$, the $\tau$-th quantile and the mean of $X_{ij}(\alpha)$. Since $\alpha$ and $\tau$ are fixed in advance, the empirical quantiles can be computed in time $O(n)$, see [34]. As there are $O(|V|^2)$ pairs, computing $W = (w_{ij})$ takes $O(|V|^2n)$. The Chu-Liu-Edmond algorithm runs on the complete bidirected graph supported by $W$, and thus takes $O(|V|^2)$, see [18]. So the complexity of QTree is $O(|V|^2n + |V|^2) = O(|V|^2n)$.

B Proof of the Consistency Theorem (Theorem 2.1)

In this section, we prove Theorem 2.1 of the Paper, which we recall here for ease of reference.

Gumbel-Gaussian noise model. For $i \in V$, the sources $Z_i$'s are i.i.d. Gumbel(1), the noises $\varepsilon_i$'s are i.i.d with symmetric, light-tailed density $f_\varepsilon$, satisfying
some weak regularity condition on its derivative formulated in (B.14), such that
\[ f_\varepsilon(x) \sim e^{-Kx^p} \text{ as } x \to \infty, \tag{B.1} \]
for some \( p > 1 \) and \( \gamma, K > 0 \).

**Theorem B.1** (Theorem 2.1 of the Paper). Assume the Gumbel-Gaussian noise model.

(a) There exists an \( r^* > 0 \) such that for any pair \( 0 < r < \tau < r^* \), the QTree algorithm with \( (w_{ij}) \) defined as the lower quantile gap
\[ w_{ij}(r, \tau) := \frac{1}{n_{ij}} \left( Q_{X_{ij}(\alpha)}(\tau) - Q_{X_{ij}(\alpha)}(r) \right)^2, \tag{B.2} \]
returns a strongly consistent estimator for the tree \( T \) as the sample size \( n \to \infty \).

(b) There exists an \( r^* > 0 \) such that for any \( 0 < r < r^* \), the QTree algorithm with \( (w_{ij}) \) defined as the quantile-to-mean gap
\[ w_{ij}(r) := \frac{1}{n_{ij}} \left( \mathbb{E}(X_{ij}(\alpha)) - Q_{X_{ij}(\alpha)}(r) \right)^2, \tag{B.3} \]
returns a strongly consistent estimator for the tree \( T \) as the sample size \( n \to \infty \).

The proof of this theorem comes in a series of steps.

As a preliminary result, Lemma B.2 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 \( T \) 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:
\[ j \sim i, \text{ that is, } i \text{ is a descendant } j \text{ in the true tree,} \]

\[ i \sim j, \text{ that is, } i \text{ is an ancestor of } j \text{ in the true tree, and} \]

\[ i \not\sim j, \text{ that is, } i \text{ is neither of the above.} \]

We first consider the case where \( W = (w_{ij}) \) is the matrix of lower quantile gaps \((B.2)\) 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 \((r, \bar{r})\) such that \( W \) is good.

Next Proposition \( B.3 \) 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 \( B.6 \) and Lemma \( B.7 \)) that among the three scenarios above, there exists some choices of quantile levels \((r, \bar{r})\) 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 \( B.9 \) playing the role of Proposition \( B.3 \).
Lemma B.2 (A criterion for ‘good’ inputs $W$). Let $W = (w_{ij})$ be a score matrix. Suppose that each true edge $j \rightarrow i \in T$ satisfies

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

and in addition, the true root $i^* \text{ satisfies}$

$$\min_{i'} w_{i'i^*} > \max_{i,j \rightarrow i} w_{ij}. \quad \text{(B.5)}$$

Then the QTree algorithm applied to input $W$ will return the true tree $T$.

Proof. The algorithm QTree applies Edmond’s algorithm to find the directed minimum spanning tree of the graph with edge weights $W$, and returns that tree. We shall prove that under the assumed conditions (B.4) and (B.5) on $W$, Edmond’s algorithm would converge after one iteration and returns the true tree $T$. Indeed, let $G$ denote the graph that consists of the smallest outgoing edge at each node. By (B.4), $G = T \cup i^* \rightarrow i'$ for some node $i' \in V$. By Edmond’s algorithm, the minimum spanning tree $T_w$ is a subset of $G$. In particular, $T_w$ is a minimum spanning tree of $G$. By (B.5), edge $i^* \rightarrow i'$ is the maximal edge. Since it belongs to the unique cycle in $G$, deleting this edge would yield the minimum directed spanning tree of $G$. Therefore $T_w = T$. \qed

B.1 Proof of Theorem 2.1 for the lower quantile gap

B.1.1 For known quantiles, $W$ is ‘good’ for appropriate choices of $(r, \tau)$

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 $(r, \tau)$. The goal is to show that there exist some appropriate choices of $(r, \tau)$ such that the resulting $W$ is ‘good’, that is, it satisfies Lemma B.2.
The first main result is Proposition B.3, 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 \( j \leadsto i \), there is a unique directed path from \( j \) to \( i \). Let \( c_{ij} \) denote the sum of all the edges along this unique path. Path uniqueness implies that \( c_{ij} = c_{ij}^* \) and \( C^* \) is transitive, i.e. \( c_{ij}^* = c_{ik}^* + c_{kj}^* \) if \( j \leadsto k \leadsto i \). Thus, by the Helmholtz decomposition on graphs \([27, equation 2.6]\), \( c_{ij}^* \) is an edge flow. That is, there exists a unique \( t^* \in \mathbb{R}^d \) with \( t^*_1 = 0 \) such that for all \( j \rightarrow i \in \mathcal{G} \),

\[
c_{ij}^* = t^*_i - t^*_j. \tag{B.6}
\]

For each \( i \in V \), define the constant

\[
\theta_i := \sum_{k \leadsto i} \exp(-t^*_k). \tag{B.7}
\]

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

\[
\xi_b := (\varepsilon_i - \varepsilon_j) + ((Z_i - Z_j) \vee b), \tag{B.8}
\]

with the convention that \( \xi_{-\infty} := (\varepsilon_i - \varepsilon_j) + (Z_i - Z_j) \). Let \( F_b \) denote the distribution function of \( \xi_b \) 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 B.3.** Assume the Gumbel-Gaussian model. Fix \( 0 < r < \tau < 1 \). Let \( w_{ij} = w_{ij}(r, \tau) \) be the lower quantile gap \([B.2]\). Fix \( j \in V \). For \( i \in V, i \neq j \), we have three cases.

1. If \( j \leadsto i \), then \( w_{ij} = q_{\tau}(F^b) - q_{\tau'}(F^b) \) for \( b = \log \theta_j - \log(\theta_i - \theta_j) \).
2. If \( j \not\leadsto i \), then \( w_{ij} = q_r(F^b) - q_{\tau'}(F^b) \) for \( b = -\infty \).
3. If \( i \leadsto j \), then \( w_{ij} = q_{1-\tau}(F^b) - q_{1-\tau'}(F^b) \) for \( b = \log \theta_i - \log(\theta_j - \theta_i) \).
Proof. It is sufficient to prove that in the noise-free case, \( w_{ij} \) equals the lower quantile gap of \((Z_i - Z_j) \lor b\). 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 \( \bar{Z}_i := Z_i - t_i^* \).

\[
\bar{X}_i = X_i - t_i^* = \bigvee_{j:j \longrightarrow i} \left(c_{ij}^* + Z_j - t_i^*\right) \quad \text{by (A.1)}
\]

\[
= \bigvee_{j:j \longrightarrow i} \left(t_i^* - t_j^* + Z_j - t_i^*\right) \quad \text{by (B.6)}
\]

\[
= \bigvee_{j:j \longrightarrow i} \bar{Z}_j. \quad \text{(B.9)}
\]

For each ordered pair \((i, j)\), define

\[
S_i = \bar{Z}_i \lor \bigvee_{i' \neq i, i' \longrightarrow i} \bar{Z}_{i'} \quad S_j = \bar{Z}_j \lor \bigvee_{j' \neq j, j' \longrightarrow j} \bar{Z}_{j'}. \quad \text{(B.10)}
\]

By definition, \( S_i \) and \( S_j \) are independent. Since \( \bar{Z}_i \)'s are translated independent Gumbel(1) by assumption, standard properties of the Gumbel(1) distribution yield that \( S_i \) and \( S_j \) are also translated independent Gumbel(1). The exact constants of translation depends 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, \( j \sim i \). Then, \( \text{(B.9)} \) implies \( \bar{X}_i = S_i \lor S_j \) and \( \bar{X}_j = S_j \). A short computation yields \( S_i \overset{d}{=} Z_i + \log(\theta_i - \theta_j) \), \( S_j \overset{d}{=} Z_j + \log \theta_j \). Therefore, denoting \( \overset{d}{=} \) equality in distribution,

\[
\bar{X}_i - \bar{X}_j = (S_i \lor S_j) - S_j = (S_i - S_j) \lor 0
\]

\[
\overset{d}{=} (Z_i - Z_j - (\log \theta_j - \log(\theta_i - \theta_j))) \lor 0
\]

\[
= ((Z_i - Z_j) \lor (\log \theta_j - \log(\theta_i - \theta_j))) - (\log \theta_j - \log(\theta_i - \theta_j))
\]

\[
= ((Z_i - Z_j) \lor b) - (\log \theta_j - \log(\theta_i - \theta_j)),
\]

41
where \( b = \log \theta_j - \log(\theta_i - \theta_j) \). Since \((\log \theta_j - \log(\theta_i - \theta_j))\) 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 \( j \leadsto i \). Computations for the third case, \( i \leadsto j \), is similar, with the role of \( i \) and \( j \) reversed, \( \tau \) is replaced by \( 1 - \tau \), and \( \tau \) is replaced by \( 1 - \tau \). For the second case, \( i \not\sim j \), then\( \bar{X}_i = S_i, \bar{X}_j = S_j \), where \( S_j \triangleq \log \theta_j + Z_j \) and \( S_i \triangleq \log \theta_i + Z_i \). Then

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

Since \((\log \theta_i - \log \theta_j)\) 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. \(\Box\)

**B.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 \( \text{B.3} \), \( w_{ij} \) is smallest when it falls in a subset of case (1), namely, \( j \rightarrow i \). We first give an overview. By Proposition \( \text{B.3} \), the lower quantile gaps \( w_{ij} \) in cases (1) and (2) are all of the form \( q(b, \tau) - q(b, \tau) \) for some constant \( b = b(i, j) \). In particular, for fixed \( j \), \( b(i, j) \) is largest when \( j \rightarrow i \). Lemma \( \text{B.5} \) says that one can choose the quantile levels \((\tau, \tau)\) such that \( q(b, \tau) - q(b, \tau) \) is monotone \textit{increasing} as a function of \( b \) on a large interval. Corollary \( \text{B.6} \) 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 \( \text{ch}(j) \).

Case (3) of Proposition \( \text{B.3} \), where \( i \) is an ancestor of \( j \), is handled by Lemma \( \text{B.7} \). The Gumbel-Gaussian assumption comes in through Lemma \( \text{B.4} \) 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 \( \eta \) is lighter than the tail of the signal differences \( Z_i - Z_j \). This is a key observation exploited in the proofs.
Lemma B.4. Under the Gumbel-Gaussian model, for any pair of nodes $i, j \in V, i \neq j$, \( \xi := Z_i - Z_j \) has density

\[ f_\xi(x) = \frac{e^x}{(1 + e^x)^2} \sim e^{-x} \text{ as } x \to \infty, \]  
(B.11)

and \( \eta := \varepsilon_i - \varepsilon_j \) has density

\[ f_\eta(x) \sim x^{1-p/2} e^{-Kx^p} \text{ as } x \to \infty. \]  
(B.12)

Proof. Computing the convolution integral yields

\[ P(Z_i - Z_j > x) = \frac{1}{1 + e^x}, \quad x \in \mathbb{R}, \]  
(B.13)

and taking the derivative gives the first statement. For the second statement, the density \( f_\varepsilon \) is a density with Gaussian tail in the sense of [5]:

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

for constants \( \gamma \) and \( \psi(x) = Kx^p \). The asymptotic form of \( f_\eta \) follows by Laplace’s integration principle as shown in [5] page 2].

We also need below the weak regularity condition that \( f_\varepsilon \) has first derivative such that

\[ f'_\eta(x) \sim f_\eta(x) \left( -Kpx^{p-1} + (1 - p/2)x^{-1} \right) \]  
(B.14)

For functions with two arguments, let \( \partial_1 \) denotes the derivative in the first argument, \( \partial_2 \) denotes the derivative in the second argument, \( \partial_{12}^2 := \partial_1 \partial_2 \) 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

\[ H(b, a) = P(\xi_b \leq a), \quad q(b, r) = r\text{-th quantile of } \xi_b. \]
Lemma B.5. Under the Gumbel-Gaussian model, for each finite constant $B$, there exists some $r^* = r^*(B) \in (0, 1)$ such that

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

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

$$q(b, \overline{r}) - q(b, \underline{r}) < q(b', \overline{r}) - q(b', \underline{r}).$$  \hspace{1cm} (B.15)

Proof. By definition,

$$H(b, q(b, r)) = r.$$  \hspace{1cm} (B.16)

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 of (B.16) with respect to $r$ gives

$$\partial_2 H \cdot \partial_2 q = 1.$$  \hspace{1cm} (B.17)

Now, differentiating both sides of (B.16) 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,

$$\partial_1 q = -\frac{\partial_1 H}{\partial_2 H}.$$  \hspace{1cm} (B.18)

Differentiate (B.17) with respect to $b$ using implicit differentiation and chain rules, we get

$$0 = \frac{\partial}{\partial b} (\partial_2 H \cdot \partial_2 q) = \frac{\partial}{\partial b} (\partial_2 H(b, q(b, r))) \cdot \partial_2 q + \partial_2 H \cdot \partial_{12}^2 q$$

$$= (\partial_{12}^2 H + \partial_{22}^2 H \cdot \partial_1 q) \cdot \partial_2 q + \partial_2 H \cdot \partial_{12}^2 q$$

$$= \frac{\partial_{12}^2 H - \partial_{22}^2 H \cdot \partial_1 H}{\partial_2 H} + \partial_2 H \cdot \partial_{12}^2 q \quad \text{by (B.17) and (B.18).}$$  \hspace{1cm} (B.19)
Rearranging the last equation gives

\[ \partial_{12}^2 q = \frac{\partial_{22}^2 H \cdot \partial_1 H - \partial_{12}^2 H \cdot \partial_2 H}{(\partial_2 H)^3}. \]  

(B.20)

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

\[ (\partial_{22}^2 H \partial_1 H - \partial_{12}^2 H \partial_2 H)(b, q(b, r)) < 0. \]  

(B.21)

Now we compute each of the terms \( \partial_2 H, \partial_1 H, \partial_{12}^2 H \) and \( \partial_{22}^2 H \) in the LHS of (B.21) explicitly in terms of the density \( f_\eta \) of the noise difference \( \eta = \varepsilon_i - \varepsilon_j \). Note that \( \xi_b = \eta + (\xi \lor b) \);

where \( \xi := Z_i - Z_j \). Then we have for \( \varepsilon > 0 \) (see Fig. (a))

\[ H(b + \varepsilon, a) - H(b, a) = \mathbb{P}(\eta + \xi \lor (b + \varepsilon) \leq a) - \mathbb{P}(\eta + \xi \lor b \leq a) \]

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

(a) \( H(b + \varepsilon, a) - H(b, a) \) is the probability that \((\xi, \eta)\) lies in the shaded regions (light + dark shaded)

(b) \( H(b, a + \varepsilon) - H(b, a) \) is the probability that \((\xi, \eta)\) lies in the shaded region (light + dark shaded)

Since \( \eta \) and \( \xi \) are independent, this implies

\[ H(b + \varepsilon, a) - H(b, a) = -\mathbb{P}(\xi \leq b)\mathbb{P}(a - b - \varepsilon \leq \eta \leq a - b) + O(\varepsilon^2). \]  

(B.22)
Hence,
\[ \partial_1 H(b, a) = \lim_{\varepsilon \downarrow 0} \frac{H(b + \varepsilon, a) - H(b, a)}{\varepsilon} = \mathbb{P}(\xi \leq b) f_\eta(a - b). \quad (B.23) \]

A similar calculation gives (see Fig. (b))
\[ \partial_2 H(b, a) = \lim_{\varepsilon \downarrow 0} \frac{H(b, a + \varepsilon) - H(b, a)}{\varepsilon} = \mathbb{P}(\xi \leq b) f_\eta(a - b) + \int_b^\infty f_\eta(a - x) f_\xi(x) \, dx \\
= : (-\partial_1 H + A)(b, a). \quad (B.24) \]

Thus,
\[ \partial_{12}^2 H(b, a) = -\mathbb{P}(\xi \leq b) f'_\eta(a - b), \]
\[ \partial_{22}^2 H(b, a) = \mathbb{P}(\xi \leq b) f'_\eta(a - b) + \int_b^\infty f'_\eta(a - x) f_\xi(x) \, dx = -\partial_{12}^2 H(b, a) + \int_b^\infty f'_\eta(a - x) f_\xi(x) \, dx = (-\partial_{12}^2 H + A_a)(b, a). \]

Now we have
\[ (\partial_{22}^2 H \partial_1 H - \partial_{12}^2 H \partial_2 H)(b, a) = (A_a \partial_1 H - A \partial_{12}^2 H)(b, a) \\
= \mathbb{P}(\xi \leq b) \left( -f_\eta(a - b) \int_b^\infty f'_\eta(a - x) f_\xi(x) \, dx + f'_\eta(a - b) \int_b^\infty f_\eta(a - x) f_\xi(x) \, dx \right). \]

Thus, \((B.21)\) holds if and only if
\[-f_\eta(q(b, r) - b) \int_b^\infty f'_\eta(q(b, r) - x) f_\xi(x) \, dx + f'_\eta(q(b, r) - b) \int_b^\infty f_\eta(q(b, r) - x) f_\xi(x) \, dx < 0. \quad (B.25)\]

Now we need to show that for each constant \(B\), there exists an \(r^*(B) > 0\) such that for all \(r < r^*\) and \(b \leq B\), \((B.25)\) holds. Fix the constant \(B\). Since the noise \(\varepsilon\) has no upper bound, \(\eta\) and hence \(\xi_b\) have unbounded support below. For each fixed \(b\), \(q(b, r) \rightarrow -\infty\) 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 \([B.4]\) to make the LHS of \((B.25)\) explicit. In particular, by \([B.12]\), as \( t \to \infty \),

\[
  f_\eta(t) = K_1 t^{1-p/2} e^{-Ktp} (1 + o(1)), \quad f'_\eta(t) = f_\eta(t) \left( -Kpt^{p-1} + (1 - p/2)t^{-1} \right) (1 + o(1)).
\]

(B.26)

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

\[
  -f_\eta(q(b,r) - b) \int_b^\infty f'_\eta(q(b,r) - x)f_\xi(x) \, dx + f'_\eta(q(b,r) - b) \int_b^\infty f_\eta(q(b,r) - x)f_\xi(x) \, dx
\]

\[
  = f_\eta(q(b,r) - b) \int_b^\infty \left[ -Kp(x - q(b,r))^{p-1} + (1 - p/2)(x - q(b,r))^{-1} \right] f_\eta(q(b,r) - x)f_\xi(x) \, dx
\]

\[
  + \left[ Kp(b - q(b,r))^{p-1} - (1 - p/2)(b - q(b,r))^{-1} \right] f_\eta(q(b,r) - b) \int_b^\infty f_\eta(q(b,r) - x)f_\xi(x) \, dx
\]

\[
  = f_\eta(q(b,r) - b) \int_b^\infty A(x)f_\eta(q(b,r) - x)f_\xi(x) \, dx,
\]

where

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

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 \( x > b \) \( \Rightarrow q(b,r) \) 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 \( (b - q(b,r))^{-1}, (x - q(b,r))^{-1} \) is smaller than \( 1 \). Thus \( \left| (1 - p/2)((b - q(b,r))^{-1} - (x - q(b,r))^{-1}) \right| \leq |1 - p/2| \). 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, \([B.25]\) holds for all \( r \in (0, r^*) \) and \( b \leq B \), as needed. \( \square \)
Corollary B.6. Under the Gumbel-Gaussian model, there exists an $r_1^* > 0$ such that: for all $0 < r < \tau < r_1^*$, 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 $b^* := \log \theta_j - \log(\theta_{ch(j)} - \theta_j)$, and let $r^*(j)$ be the constant $r^*$ that works for $B = b^*$ in Lemma B.5. By Proposition B.3

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

Now we consider two cases.

Case 1: $i'$ is a descendant of $j$, that is, $j \rightsquigarrow i'$. Then by Proposition B.3

$$w_{j,i'} = q(b,\tau) - q(b,\tau)$$

where $b = \log \theta_j - \log(\theta_{i'} - \theta_j)$. But since $i' \neq ch(j)$, $i'$ must be a descendant of $i$ as well. By definition of $\theta$’s in (B.7), $i \rightsquigarrow i'$ implies $\theta_{i'} > \theta_i$. Therefore, $b < b^*$, so by (B.15), $w_{ij} < w_{i',j}$. This concludes case 1.

Case 2: $j \not\rightsquigarrow i'$. Then by Proposition B.3

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

Since $-\infty < b^*$, so by (B.15), $w_{ij} < w_{i',j}$. This concludes case 2. \hfill \square

Lemma B.7. There exists some $r_2^* > 0$ such that for all $0 < \tau < \tau < r_2^*$, for all $j \in V, i' \rightsquigarrow j$ implies

$$q(b,\tau) - q(b,\tau) < q(b',1-\tau) - q(b',1-\tau),$$  \hspace{1cm} (B.27)
where \( b = \log \theta_j - \log(\theta_{ch(j)} - \theta_j) \) and \( b' = \log \theta_i - \log(\theta_j - \theta_i') \). In particular, if \( i' \sim j \), then for all quantile levels \( r, \tau \) such that \( 0 < r < \tau < r^*_2 \),

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

**Proof.** It is sufficient to prove that (B.27) 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 (B.27) to relate it to the partial derivatives of \( H \). Define

\[
\mathcal{B} := \{ \log \theta_j - \log(\theta_i - \theta_j) : i, j \in V, j \sim i \}. \quad (B.28)
\]

Note that (B.27) is equivalent to

\[
\partial_2 q(b, r) < \partial_2 q(b', 1 - r) \quad \text{for all } r \in (0, r^*_2) \text{ and for all } b, b' \in \mathcal{B}. \quad (B.29)
\]

By (B.17), 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 (B.24), \( \partial_2 H > 0 \) point-wise, thus our goal now is to show that for sufficiently small \( r \),

\[
\partial_2 H(b', q(b', 1 - r)) - \partial_2 H(b, q(b, r)) < 0 \quad (B.30)
\]

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 \( f_\eta \) and \( f_\xi \) using (B.24), then apply Lemma B.4. Indeed, by (B.24),

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

By Lemma B.4, \( f_\xi \) has heavier tail than \( f_\eta \), 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

\[
\partial_2 H(b', a) > b_1 f_\xi(a). \quad (B.31)
\]
Now we consider $\partial_2 H(b, -a)$. From (B.24),
\[
\partial_2 H(b, -a) = \mathbb{P}(\xi \leq b)f_\eta(-a-b) + \int_b^\infty f_\eta(-a-x)f_\xi(x) \, dx.
\]
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 (B.31) and noting that $\partial_2 H(b, a)$ is just the density $f_{\xi_b}(a)$ of $\xi_b$, we get
\[
f_{\xi_b}(-a) = O(f_{\xi_{b'}}(a)) \tag{B.32}
\]
for all $b, b' \in \mathcal{B}$ and $a$ large. Now $\partial_2 H(b, \eta(b, r))$ is just the slope of the cdf of $f_{\xi_b}$ at its $r$-th quantile. Therefore, for $r$ small, by (B.32), $\partial_2 H(b', \eta(b', 1-r)) < \partial_2 H(b, \eta(b, r))$ which proves (B.30) and thus completes the proof of (B.27). The last statement follows from Proposition B.3, case (3).

**Corollary B.8.** If the true quantiles are known, then there exist some choices of $(r, \tau)$ such that the lower quantile gap matrix $W$ satisfies the conditions of Lemma B.2, that is, (B.4) and (B.5).

**Proof.** Set $r^* = \min(r_1^*, r_2^*)$ where $r_1^*$ origins from Corollary B.6 and $r_2^*$ origins from Lemma B.7. Let $(r, \tau)$ be any pair such that $0 < r < \tau < r^*$, and let $W$ be the corresponding lower quantile gap matrix with the true quantiles. Lemma B.7 guarantees that
is satisfied for $W$. Then, Corollary B.6 and Lemma B.7 together say that (B.4) is satisfied for $W$. □

**Proof of Theorem 2.1 for the lower quantile gap**

Fix $(r, \tau)$ such that Corollary B.8 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 B.2, 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 B.8, $W$ 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 B.2 □

**B.2 Proof of Theorem 2.1 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 B.3 is the following.

**Proposition B.9.** Fix $r \in [0, 1)$, and let $w_{ij}$ be the quantile-to-mean gap (B.3). Assume the Gumbel-Gaussian model. Then

1. If $j \sim i$, then $w_{ij} = \log \theta_i - \log(\theta_i - \theta_j) - q_r(\xi^b)$ where $b = \log \theta_j - \log(\theta_i - \theta_j)$. 51
(2) If \( j \not\sim i \), then \( w_{ij} = -q_b(\xi^b) \) where \( b = -\infty \).

(3) If \( i \leadsto j \), then \( w_{ij} = q_1(\xi^b) - (\log \theta_j - \log(\theta_i - \theta_j)) \) where \( b = \log \theta_i - \log(\theta_j - \theta_i) \).

Instead of a lengthy proof of the analog of Proposition B.3 by duplicating arguments, we provide some informal reasoning. We check that our quantile-to-mean gaps \( w_{ij} \) satisfy the inequalities of Corollary B.6 and Lemma B.7 by first checking the noise-free case, where \( \varepsilon_i \equiv \varepsilon_j \equiv 0 \). We consider the three cases of Proposition B.9.

1. If \( j \leadsto i \). Then \( \xi^b \) has a left-most atom at \( b = \log \theta_j - \log(\theta_i - \theta_j) \), so for sufficiently small \( r \), \( w_{ij} = \theta_i - \theta_j \). This is minimal when \( i \) is a direct descendant of \( j \). So Corollary B.6 for the case \( j \leadsto i \) holds in the noise-free case.

2. If \( j \not\sim i \). Then \( \xi^b \) has no left-most atom, so as \( r \downarrow 0 \), \( q_r(\xi^b) \to -\infty \), so \( w_{ij} \to \infty \). So Corollary B.6 also holds in the noise-free case for the remaining case, \( j \not\sim i \).

3. If \( i \leadsto j \). Then \( \xi^b \) has a left-most atom, but no right-most atom. Again, as \( r \downarrow 0 \), \( q_1(\xi^b) \to \infty \), so \( w_{ij} \to \infty \). Thus, Lemma B.7 holds in the noise-free case.

Now we consider the effect of noise. We send \( r \downarrow 0 \). As long as \( \eta := \varepsilon_i - \varepsilon_j \) has lighter tail than \( Z_i - Z_j \), as guaranteed by Lemma B.4, then we have the following.

- In case (1), \( q_b(\xi^b) \) is dominated by the lower tail of \( \eta \).
- In case (2), \( q_b(\xi^b) \) 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), \( q_1(\xi^b) \) is dominated by the upper tail of \( Z_i - Z_j \), and in particular, is going to \( \infty \) at a faster rate than case (1).
This domination calculation is the same calculation done in the proof of Lemma B.7. The above says that for fixed $j$, for small enough $r$, 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 $\xi'_{ij} = (Z_i - Z_j) \lor (\log \theta_j - \log(\theta_i - \theta_j))$. For fixed $j$, $(\xi'_{ij} : j \leadsto i)$ is a particular family of distribution indexed by $i$. By a decoupling argument, it is sufficient to show that $q_x(\xi'_{ij})$ 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 2.1 for the quantile-to-mean gap.

## C Further details on the simulation study

Our main result, Theorem 2.1 ensures strong consistency of the output trees of the QTree algorithm when the sample size $n \to \infty$. In this section we perform a small simulation study to show that QTree even works well for a relatively small sample size $n$.

We generate data $\mathcal{X}$ from a max-linear Bayesian tree with $|V| = d$ nodes as defined in equation (2.2) of the Paper, where the sources $Z_1, \ldots, Z_d$ are independent Gumbel(1) distributed. For each node $i$, we calculate the sample standard deviation $\hat{\sigma}_{X_i}$ of $(X^1_i, \ldots, X^n_i)$ and take the sample median $\hat{\sigma}$ over all nodes $1, \ldots, d$. We then generate i.i.d. normally distributed noise variables $\varepsilon^t_i$ with mean zero and standard deviation $k \cdot \hat{\sigma}$ for $i \in V$ and $t = 1, \ldots, n$. For the noise-to-signal ratio $k$, we choose $k = 30\%$.

We generate a random undirected spanning of size $d$ using the Graph generators module of networkx in Python. We then choose the root node uniformly at random which uniquely
determines the root-directed spanning tree. Finally, we assign edge weights independently. For every edge, we draw an edge weight $c_{ij}$ from the interval $[\log(0.2), \log(1)]$ uniformly.

For the score, we take the monotone transformation of the quantile-to-mean gap as in (B.2) given by

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

and apply the QTree Algorithm [1] from the Paper with parameters $r = 0.05$ and $\alpha = 0$. A sensitivity analysis has shown that altering $r$ influences the results only minorly. We use graph size $d \in \{10, 30, 50, 100\}$ 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 (1.1) or [45]. 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.

In Figures (13a)–(13d) and Figures (14a)–(14d) we depict the mean nSHD and TPR for all four graph sizes. Both metrics quickly drop to zero, respectively one, as the sample sizes $n$ increase. Moreover, comparing the four figures for a fixed sample size $n$, the metrics perform only slightly worse for increasing graph size $d$. The decrease in performance can be seen in Figure [15] where we plot the minimum amount of data $n$ needed to reach a mean nSHD of 5%. Since larger networks have more opportunities for a wrong causal influence, this is in line with what we expect.
Figure 13: Mean nSHD for different graph sizes $d$ 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 $n \to \infty$. Moreover, the metric tends to converge faster for $(\mathcal{R}, \hat{\mathcal{R}})$ suggesting that the majority of wrong edges are still stream-connected. Increasing the graph size only minorly decreases the performance.
Figure 14: Mean TPR for different graph sizes $d$ and noise-to-signal ratio $k = 30\%$. Solid lines denote the metric for the pair $(T, \hat{T})$ while dashed lines denote the metric for its reachability pair $(R, \hat{R})$. For all graph sizes, the TPR quickly converges to 1 as $n \to \infty$. Moreover, the metric tends to converge faster for $(R, \hat{R})$ suggesting that the majority of wrong edges are still stream-connected. Increasing the graph size only minorly decreases the performance.
Moreover, the nSHD of $(\hat{R}, R)$ is generally lower than the nSHD of $(\hat{T}, T)$ for larger $n$, which implies that the algorithm tends to connect stream-connected nodes leading to less severe mistakes. Overall, QTree performs reliably even for very large networks and small sample sizes. For example, even with a strong noise-to-signal ratio of 30% and a sample size of only $n = 100$, we achieve a mean nSHD of around $6 - 20\%$, depending on the size of the graph. For this reason, we conclude that QTree works very well even for small sample sizes.

**Figure 15:** Minimum amount of observations needed to reach a mean nSHD of 5%. For increasing graph size $d$, the amount of data needed only increases minorly to reach the same level of performance.
D Supplemental Figures

Figure 16: Metrics nSHD, TPR, FDR and FPR for the Top Colorado and varying parameters $\alpha$. For detailed explanations see Figure 10 of the Paper.
Figure 17: Metrics nSHD, TPR, FDR and FPR for the Middle Colorado and varying parameters $\alpha$.
For detailed explanations see Figure 10 of the Paper.
Figure 18: Metrics nSHD, TPR, FDR and FPR for the trimmed Bottom Colorado (Bottom150) and varying parameters $\alpha$. For detailed explanations see Figure 10 of the Paper.