

A Relationship to contribution analysis

We first sketch the causal framework adopted by Budhathoki et al. [23]. In particular, they assume that the causal relationships between X_1, \dots, X_n are described by a Structural Causal Model (SCM). In an SCM, each variable X_i is a function f_i of its parents PA_i in the causal graph, and an unobserved noise term N_i :

$$X_i := f_i(PA_i, N_i), \quad (14)$$

where the noise variables N_1, \dots, N_n are jointly independent [33]. In addition, it is assumed that the SCM is invertible [49]. That is, the noise value n_j of N_j can be recovered from the value x_j of its corresponding observed variable X_j and the values pa_j of its parents.

As in our case, the value x_n is flagged as an anomaly, and we wish to identify its root cause from among the variables (X_1, \dots, X_n) . Treating (without loss of generality) X_n as a sink node in the causal DAG, iterative application of Eq. 14 results in the representation

$$X_n = F(N_1, \dots, N_n), \quad (15)$$

in which the structural information is implicit in the function F . By Eq. 15 we therefore have that $x_n = F(n_1, \dots, n_n)$ where $(n_1, \dots, n_n) := \mathbf{n}$ denote the corresponding values of the noise variables for sample (x_1, \dots, x_n) . This representation makes clear that a node X_j 's contribution to the anomaly x_n is ultimately attributable to the contribution of its noise term [23, 50].

If you consider our presentation of RCA, wherein we attribute the anomaly x_n to one of the causal mechanisms being corrupted, the characterisation in terms of noise terms may appear quite different. However, we can connect the two perspectives by noting that one can think of each value n_j as randomly switching between deterministic mechanisms $f_j(\cdot, n_j)$ – so-called *response functions* [51]. If a noise term is "corrupted" and takes an unusual value n_j , the corresponding mechanism generating x_j from its parents pa_j will be unusual as well. The goal of finding which causal mechanism did not work as expected, then is common between the two perspectives. The intuition behind the approach of Budhathoki et al. is that if for the root cause X_j the unusual value of its noise term n_j were replaced by a "normal" one, then x_n would change to a non-anomaly with high probability. This gives the basis for defining how much each variable contributed to the observed anomaly. We explain how this is formalised next.

A.1 Contribution analysis

To compute the contribution of each noise N_i to the anomaly x_n , Budhathoki et al. [23] measure how replacing the observed value n_i (originating from a potentially corrupted mechanism) with a random "normal" value, sampled from its regular distribution $P(N_i)$, changes the likelihood of the anomaly event, $E = \{\tau(X_n) \geq \tau(x_n)\}$. Intuitively, this shows us the extent to which n_i was responsible for the extremeness of x_n .

Note, however, that the influence of the value n_i on the likelihood of the anomaly event will also depend on the values of the other noise variables. In order to assess the contribution of n_i we therefore also need to consider how it depends on the context, i.e. how it changes depending on which other noise variables we similarly replace with random "normal" values. To formalise this, for any subset of the index set, $S \subseteq \mathcal{U} := \{1, \dots, n\}$, first note that the probability of the anomaly event when all nodes in S are set to their observed value and all the nodes in $\bar{S} = \mathcal{U} \setminus S$ are randomised is $P(E | \mathbf{N}_S = \mathbf{n}_S)$. Now the contribution of a node $j \notin S$ given the context S is defined as:⁷

$$C(j|S) := \log \frac{P(E | \mathbf{N}_S = \mathbf{n}_S, \mathbf{N}_j = \mathbf{n}_j)}{P(E | \mathbf{N}_S = \mathbf{n}_S)}. \quad (16)$$

To give a "fair" attribution among the noise variables, the authors adopt Shapley values. Let $\Pi : \mathcal{U} \rightarrow \mathcal{U}$ be the set of all possible permutations of the nodes and $\pi \in \Pi$ be any permutation. One then defines the contribution of a node j given permutation π as $C^\pi(j) := C(j | I^{\pi < j})$, where $I^{\pi < j}$ denotes the set of nodes that appear *before* j with respect to π , i.e., $I^{\pi < j} = \{i \in \mathcal{U} \mid \pi(i) < \pi(j)\}$. We easily see that for each permutation π , $S(x_n)$ decomposes into the contributions of each node,

⁷Note the difference in the definition of $C(\cdot | S)$ compared to [23]. In our case S is the set for which $\mathbf{N}_S = \mathbf{n}_S$ while in [23] $\mathbf{N}_{\mathcal{U} \setminus S} = \mathbf{n}_{\mathcal{U} \setminus S}$.

i.e., $S(x_n) = \sum_{j \in \{1, \dots, n\}} C^\pi(j)$. The Shapley contribution of a node is then given by averaging over all the possible permutations in Π :

$$C^{Sh}(j) := \frac{1}{n!} \sum_{\pi \in \Pi} C(j | I^{\pi < j}). \quad (17)$$

This approach therefore provides a full quantitative contribution analysis of root causes. However, it suffers from some practical issues. Firstly, the Shapley contribution of a node is expensive to compute [23]. More fundamentally, however, for most permutations π , the contributions $C^\pi(j)$ rely on knowing the structural equations [14]. As the SCM cannot generally be inferred even with interventional data, this is a serious bottleneck for the application of this approach. As the approaches we propose in this paper do not depend on knowing the structural equations (or even the causal graph in the case of SCORE ORDERING), we explain next how we can nonetheless recast our results in terms of the contribution approach just presented.

A.2 Interventional vs counterfactual RCA

The key result we will need is that so long as π is a topological ordering of the nodes in the causal graph, then all contributions $C^\pi(j)$ do not require knowing the SCM. To illustrate this, first consider the bivariate case $X \rightarrow Y$ with structural equations $X := N_X$ and $Y := f(X, N_Y)$, and anomaly y in sample (x, y) . To determine the contribution of N_Y to the anomaly, we consider randomising N_Y and fixing $N_X = n_X = x$. This generates Y according to the observational distribution $P(Y | x)$, so its estimation does not require knowledge of the SCM. On the other hand, randomising N_X and fixing $N_Y = n_Y$ cannot be resolved into any observational term (see also Section 5 in [52]).⁸

The following result generalises this insight:

Proposition A.1. *Whenever π is a topological order of the causal DAG, i.e., there are no arrows $X_i \rightarrow X_j$ for $\pi(i) > \pi(j)$, all contributions $C^\pi(j)$ can be computed from observational conditional distributions.*

This will allow us to connect our approach to that of Budhathoki et al. [23] in the following sense: if our goal is to simply identify a single root cause, then as long as it is "dominating" in the sense that it has the largest contribution of any variable regardless of the order chosen, Proposition A.1 says we can identify it without knowing the SCM.

It will prove instructive to generalise the notion of the contribution of a *single node* in [16] to the contribution of a *set* $\mathcal{R} \subseteq \mathcal{U} \setminus \mathcal{S}$ of nodes, given the context \mathcal{S} , i.e.,

$$C(\mathcal{R} | \mathcal{S}) := \log \frac{P(E | \mathbf{N}_{\mathcal{R}} = \mathbf{n}_{\mathcal{R}}, \mathbf{N}_{\mathcal{S}} = \mathbf{n}_{\mathcal{S}})}{P(E | \mathbf{N}_{\mathcal{S}} = \mathbf{n}_{\mathcal{S}})}. \quad (18)$$

This notion becomes rather intuitive after observing that it is given by the sum of the contributions of all the elements in \mathcal{R} when they are one by one added to the context \mathcal{S} :

Lemma A.2. *For any set $\mathcal{R} \subseteq \mathcal{U} \setminus \mathcal{S}$, it holds that $C(\mathcal{R} | \mathcal{S}) := C(j_1 | \mathcal{S}) + \sum_{i=2}^k C(j_i | \mathcal{S} \cup \{j_1, \dots, j_{i-1}\})$ with $\mathcal{R} = \{j_1, \dots, j_k\}$.*

Next, the following result shows that a set of nodes are unlikely to obtain a high contribution when the noise values are randomly drawn from their usual distributions, i.e., we have the following proposition:

Proposition A.3. *Whenever all noise variable in some set \mathcal{R} are sampled from $P(\mathbf{N}_{\mathcal{R}})$, it holds that $P(C(\mathcal{R} | \mathcal{S}) \geq \alpha) \leq e^{-\alpha}$.*

Note that the proposition allows that the variables N_j not in \mathcal{R} are drawn from a different distribution. Thus, Proposition A.3 states that it is unlikely that a set that only contains *non-corrupted nodes* has high contribution.

Next, we will show how our main results can be recast in terms of bounds on contributions, rather than on p -values.

⁸This can be checked by an SCM with two binaries where $Y := X \oplus N_Y$, with unbiased N_Y , which induces the same distribution as the SCM $Y := N_Y$, where X and Y are disconnected and thus X has a contribution of zero.

A.3 Bounds on contributions

Consider, for illustration, the setting discussed in subsection 3.1 wherein the causal DAG is $X \rightarrow Y$, we observe (x, y) with anomaly scores $S(x), S(y)$, and we wish to find the root cause of anomaly y . Under the same assumptions as stated there we have the following proposition:

Proposition A.4. *Subject to score typicality, the contributions of y and x on the anomaly y satisfy:*

$$C(x) \leq S(x) \quad \text{and} \quad C(y | x) \geq |S(y) - S(x)|_+$$

Proof. Proposition A.1 permits rewriting contributions in terms of observational probabilities, so that, together with score typicality, we have:

$$\begin{aligned} C(y | x) &:= \log \frac{P(\tau(Y) \geq \tau(y) | Y = y, X = x)}{P(\tau(Y) \geq \tau(y) | X = x)} \\ &= \log \frac{1}{P(\tau(Y) \geq \tau(y) | X = x)} \\ &= S(y | x) \geq |S(y) - S(x)|_+, \end{aligned}$$

and using $C(x) + C(y | x) = S(y)$ we obtain the bound for $C(x)$. \square

We can now further interpret the conclusions drawn in subsection 3.1 in terms of contributions. For example, suppose we have $S(y) \gg S(x)$. In the case that $X \rightarrow Y$, Proposition A.4 says that y must have a large contribution to the anomaly, while x must have a smaller one. We would therefore favour y as the root cause. Alternatively, if $S(x) \gg S(y)$, then we cannot conclude that y has a large contribution, but we would reject the hypothesis that the mechanism generating x worked as normal (from Lemma 3.2), and its contribution may be large. Under the working hypothesis that there is only a single root cause, we would favour x . Following the extensions from the bivariate case in subsections 3.2 and 3.3, we can similarly recast the results in terms of contributions as above. Naturally, this does not alter any of the conclusions, but rather demonstrates that our results do not depend strictly on our adopted formalisation of the RCA problem in terms of CBNs, but additionally admits an interpretation in terms of contributions.

B Sample complexity of information-theoretic anomaly score estimation

To reliably tell whether the IT anomaly score of one anomaly exceeds that of another, we need to estimate each score up to an error of at least half their difference. We investigate how many samples are required to estimate the scores $S(x_1), \dots, S(x_n)$ up to an error level δ :

Lemma B.1. *If we use at least*

$$k = \frac{3e^{S_{\max}}}{\delta^2} \log \left(\frac{2n}{\alpha} \right) \quad (19)$$

samples from the ‘normal’ period, the score estimates $\hat{S}(x_1), \dots, \hat{S}(x_n)$ using the estimator in Equation (5) satisfy $|\hat{S}(x_i) - S(x_i)| < \delta$ for all $i \in \{1, \dots, n\}$ with probability at least $1 - \alpha$.

Proof. Let $p_i := P(\tau(X_i) \geq \tau(x_i))$ and estimate $\hat{p}_i := \frac{1}{k} \sum_{j=1}^k \mathbf{1}\{\tau(x_i^j) \geq \tau(x_i)\}$ so that $S(x_i) = -\log p_i$ and $\hat{S}(x_i) = -\log \hat{p}_i$ as in Equations (4) and (5), respectively. $|\hat{S}(x_i) - S(x_i)| < \delta$ is equivalent to,

$$|\log \hat{p}_i - \log p_i| < \delta \implies e^{-\delta} < \frac{\hat{p}_i}{p_i} < e^{\delta}. \quad (20)$$

Rearranging, we have $p_i(e^{-\delta} - 1) < \hat{p}_i - p_i < p_i(e^{\delta} - 1)$, and noting $e^{\delta} - 1 > 1 - e^{-\delta}$ for all $\delta > 0$, we have

$$|\hat{p}_i - p_i| < \epsilon_i := (e^{\delta} - 1)p_i, \quad (21)$$

so that when δ is small we have $\epsilon_i \approx \delta p_i$. We can then use the multiplicative form of the Chernoff bound for $0 \leq \delta \leq 1$ to say,

$$P(|\hat{p}_i - p_i| \geq \delta p_i) \leq 2 \exp \left(-\frac{\delta^2 p_i k}{3} \right). \quad (22)$$

Noting that $\min_i p_i = \min_i e^{-S(x_i)} = e^{-\max_i S(x_i)} =: e^{-S_{\max}}$ and applying the union bound,

$$P\left(\bigcup_{i=1}^n \{|\hat{p}_i - p_i| \geq \delta p_i\}\right) \leq 2n \exp\left(-\frac{\delta^2 e^{-S_{\max}} k}{3}\right), \quad (23)$$

so that finally,

$$2n \exp\left(-\frac{\delta^2 e^{-S_{\max}} k}{3}\right) < \alpha \implies k > \frac{3e^{S_{\max}}}{\delta^2} \log\left(\frac{2n}{\alpha}\right)$$

□

C Proofs

C.1 Proof of Lemma 3.4

Since we assume that $S(X)$ is a continuous variable with density with respect to the Lebesgue measure, it follows from the properties of IT scores that it is distributed according to the density $p(S(X) = s) = e^{-s}$ for $s \geq 0$. Accordingly, the conditional distribution of $S(X)$, given $S(X) \in [s_X, S(y)]$ has the density $p(S(X) = s | S(X) \in [s_X, S(y)]) = e^{-s}/(e^{s_X} - e^{-S(y)})$. We want to compare averages of the expressions

$$P(S(Y) \geq S(y) | S(X) = s) \quad \text{versus} \quad e^{-|S(y) - s|_+}$$

over the interval $[s_X, S(y)]$. Averaging the difference of the two expressions with respect to the above-mentioned density yields

$$\begin{aligned} & \int_{s_X}^{S(y)} \left(P(S(Y) \geq S(y) | S(X) = s) - e^{-|S(y) - s|_+} \right) p(s | S(y) \geq S(X) \geq s_X) ds \\ &= P(S(Y) \geq S(y) | S(y) \geq S(X) \geq s_X) - \int_{s_X}^{S(y)} e^{s-S(y)} \frac{e^{-s}}{e^{-s_X} - e^{-S(y)}} ds \\ &\leq \frac{e^{-S(y)}}{e^{-s_X} - e^{-S(y)}} - \int_{s_X}^{S(y)} \frac{e^{-S(y)}}{e^{-s_X} - e^{-S(y)}} ds = \frac{e^{-S(y)}}{e^{-s_X} - e^{-S(y)}} (1 - S(y) + s_X) \leq 0. \end{aligned}$$

The first inequality holds because $P(B|A) \leq P(B)/P(A)$ for any two events A, B and the second because we have assumed $s_X \leq S(y) - 1$.

Applying Markov's inequality, we therefore have

$$P_{X|S(X) \in [s_X, S(y)]} [P(S(Y) \geq S(y) | S(X)) \geq c \cdot e^{-|S(y) - S(X)|_+}] \leq 1/c. \quad (24)$$

As X may in general be multivariate, S will not in general be injective, and so we cannot substitute conditioning on a particular value of the score for conditioning on the associated x , which is required to give a bound on the conditional score. As such, note that

$$P(S(Y) \geq S(y) | S(X) = s) = \int P(S(Y) \geq S(y) | X = x) p(x | S(X) = s) dx,$$

so that applying Markov's inequality for any given s ,

$$P_{X|S(X)=s} [P(S(Y) \geq S(y) | X) \geq c \cdot P(S(Y) \geq S(y) | S(X))] \leq 1/c. \quad (25)$$

As the inequality holds for all s , we can take the average of the LHS with respect to $p(S(X) = s | S(X) \in [s_X, S(y)])$ to say,

$$P_{X|S(X) \in [s_X, S(y)]} [P(S(Y) \geq S(y) | X) \geq c \cdot P(S(Y) \geq S(y) | S(X))] \leq 1/c. \quad (26)$$

Applying the union bound to the events in equations 24 and 26, the joint event

$$\begin{aligned} P(S(Y) \geq S(y) | S(X)) &\leq c \cdot e^{-|S(y) - S(X)|_+} \quad \text{and} \\ P(S(Y) \geq S(y) | X) &\leq c \cdot P(S(Y) \geq S(y) | S(X)) \end{aligned}$$

occurs with probability at least $1 - 2/c$, and in such case the following inequality holds:

$$P(S(Y) \geq S(y) | X = x) \leq c^2 \cdot e^{-|S(y) - S(x)|_+}. \quad (27)$$

Finally, noting we can exchange $S(Y) \geq S(y)$ for $\tau(Y) \geq \tau(y)$ due to the properties of IT scores, and taking the negative logarithm of both sides, for x sampled at random according to $P(X = x | S(X) \in [s_X, S(y)])$,

$$S(y | x) \geq |S(y) - S(x)|_+ - 2 \log c, \quad (28)$$

holds with probability at least $1 - 2/c$, as desired.

C.2 Proof of Lemma 3.5

$$\begin{aligned} P(\tau(Y) \geq \tau(y) \mid X = x) &= P(S(Y) \geq S(y) \mid X = x) = P(S(Y) \geq S(y) \mid S(X) = S(x)) \\ &\leq P(S(Y) \geq S(y) \mid S(X) \geq S(x)) \leq \frac{P(S(Y) \geq S(y))}{P(S(X) \geq S(x))} = \frac{e^{-S(y)}}{e^{-S(x)}}. \end{aligned}$$

Where the first equality follows from the definition of IT scores and the second from injectivity. The first inequality follows from monotonicity, and the second because $P(B \mid A) \leq P(B)/P(A)$ for any two events A, B . The final equality again follows from the definition of IT scores. Taking the negative logarithm of both sides, and remembering that anomaly scores are non-negative, we prove the lemma.

C.3 Proof of Lemma 3.7

Assume without loss of generality that X_1, \dots, X_n is a topological ordering of the DAG. Then $S(X_i \mid X_1, \dots, X_{i-1}) = S(X_i \mid PA_i)$ holds due to the local Markov condition. Since $S(X_j \mid PA_j = pa_j)$ has density e^{-s} for all pa_j , also $S(X_i \mid X_1 = x_1, \dots, X_{i-1} = x_{i-1})$ has density e^{-s} for all x_1, \dots, x_{i-1} . Hence, $S(X_i \mid PA_i)$ is independent of X_1, \dots, X_{i-1} .

C.4 Proof of Lemma 3.8

From Lemma 3.7 and the properties of IT scores, if H_0^j holds then $S(X_i \mid PA_i)$ for $i \neq j$ are independent, standard Exponential random variables. The sum of $n - 1$ independent, standard Exponential random variables is Erlang distributed [53] with CDF

$$F(x) = 1 - e^{-x} \sum_{l=0}^{n-2} \frac{x^l}{l!},$$

and as such, S_{sum} is Erlang distributed. By the probability integral transform $U := F(S_{\text{sum}})$ is uniformly distributed on $[0, 1]$, and so by symmetry, so is $1 - U$. Again by the probability integral transform, $-\log(1 - U)$ is standard Exponential, so finally noting that $S = -\log(1 - F(S_{\text{sum}}))$, we have the result.

C.5 Proof of Theorem 3.9

The theorem is a direct result of Lemma 3.7 and Lemma 3.8

C.6 Proof of Theorem 3.11

$$s_{\text{sum}} := \sum_{i \neq j} S(x_i \mid pa_i) \geq \sum_{i \neq j} |S(x_i) - S(pa_i)|_+ =: \hat{s}_{\text{sum}}, \quad (29)$$

where the inequality follows from score typicality of event (x_1, \dots, x_n) , and s_{sum} and \hat{s}_{sum} are the realisations of S_{sum} and \hat{S}_{sum} , respectively. The theorem then follows directly from Theorem 3.9 with \hat{S}_{sum} in place of S_{sum} .

C.7 Proof of Theorem 3.12

From Lemma 3.7 we have that the conditional anomaly score for the anomaly with the maximum score gap must be at least δ_{max} . From Lemma 3.4, and the properties of IT anomaly scores, the conditional anomaly scores are independent and each distributed according to density $p(s) = e^{-s}$. Under H_0^{max} , assume without loss of generality that x_n corresponds to the true root cause. Then the remaining $(n - 1)$ conditional scores are i.i.d. $\text{Exp}(1)$, and

$$P(S(X_i \mid PA_i) < \delta \text{ for all } i < n) = (P(S(X_i \mid PA_i) < \delta))^{n-1} = (1 - e^{-\delta})^{n-1}.$$

Therefore, the probability of observing at least one conditional anomaly score of at least δ_{max} among the $(n - 1)$ non-root causes is $1 - (1 - e^{-\delta_{\text{max}}})^{n-1}$. This probability upper bounds the probability of observing at least one score gap of δ_{max} or greater as score gaps lower bound conditional scores.

C.8 Proof of Theorem 3.13

The proof of the theorem is already almost complete, following the argument in the main text. As noted there, $H_0^{\text{top-}k}$ implies that along the causal path the score of $x_{\pi(1)}$ must be higher than an ancestral node by at least Δ_k . Lemma 3.3 along with the coarsening argument, implies that, individually, the hypothesis that the mechanism generating x worked as expected, where $S(x)$ is higher than the score of an ancestral node by Δ_k , can be rejected at level $p \leq e^{-\Delta_k}$. Therefore, along the causal path, the probability that the score of *any* of the (maximum of) n nodes exceeds an ancestral node by Δ_k is at most $n \cdot e^{-\Delta_k}$, applying the union bound. However, note that we need to reapply this argument for each of the chains incident on $X_{\pi(1)}$ in the graph. The number of incident chains is at most d_{\max} , and so again applying the union bound, we arrive at the final bound given in the theorem.

C.9 Proof of Proposition A.1

With the event E as in Eq. 3 and $C^\pi(j) := C(j|I^{\pi < j})$ we have

$$\begin{aligned} C^\pi(j) &= \log \frac{P(E|\mathbf{N}_{\pi < j \cup \{j\}} = \mathbf{n}_{\pi < j \cup \{j\}})}{P(E|\mathbf{N}_{\pi < j} = \mathbf{n}_{\pi < j})} \stackrel{i}{=} \log \frac{P(E|\mathbf{X}_{\pi < j \cup \{j\}} = \mathbf{x}_{\pi < j \cup \{j\}})}{P(E|\mathbf{X}_{\pi < j} = \mathbf{x}_{\pi < j})} \\ &\stackrel{ii}{=} \log \frac{P(E|do(\mathbf{X}_{\pi < j \cup \{j\}} = \mathbf{x}_{\pi < j \cup \{j\}}))}{P(E|do(\mathbf{X}_{\pi < j} = \mathbf{x}_{\pi < j}))}. \end{aligned} \quad (30)$$

(i) and (ii) are seen as follows:

$$P(E|\mathbf{N}_{\mathcal{I}}) = P(E|\mathbf{X}_{\mathcal{I}}) = P(E|do(\mathbf{X}_{\mathcal{I}})). \quad (31)$$

The first equality in Eq. 31 follows from $X_n \perp X_{\mathcal{I}} | \mathbf{N}_{\mathcal{I}}$ and because $\mathbf{X}_{\mathcal{I}}$ is a function of $\mathbf{N}_{\mathcal{I}}$. The second one follows because conditioning on all ancestors blocks all backdoor paths. Note that since π is a topological ordering of the nodes, all $\pi < j$ are ancestors of j .

C.10 Proof of Lemma A.2

Denote $q(\mathcal{S}) = P(E | \mathbf{N}_{\mathcal{S}} = \mathbf{n}_{\mathcal{S}})$ then we have that

$$\begin{aligned} C(\mathcal{R}|\mathcal{S}) &= \log q(\mathcal{S} \cup \mathcal{R}) - \log q(\mathcal{S}) \\ &= (\log q(\mathcal{S} \cup \mathcal{R}) - \log q(\mathcal{S} \cup \mathcal{R} \setminus \{j_k\})) + \dots \\ &\quad (\log q(\mathcal{S} \cup \mathcal{R} \setminus \{j_k\}) - \log q(\mathcal{S} \cup \mathcal{R} \setminus \{j_k, j_{k-1}\})) + \dots \\ &\quad + (\log q(\mathcal{S} \cup \{j_1\}) - \log q(\mathcal{S})) = C(j_1|\mathcal{S}) + \sum_{i=2}^k C(j_i|\mathcal{S} \cup \{j_1, \dots, j_{i-1}\}). \end{aligned}$$

C.11 Proof of Proposition A.3

By definition, we have that

$$C(\mathcal{R}|\mathcal{S}) = \log \frac{P(E|\mathbf{N}_{\mathcal{R}} = \mathbf{n}_{\mathcal{R}}, \mathbf{N}_{\mathcal{S}} = \mathbf{n}_{\mathcal{S}})}{P(E|\mathbf{N}_{\mathcal{S}} = \mathbf{n}_{\mathcal{S}})},$$

and we use $P(E|\mathbf{n}_{\mathcal{S}})$ instead of $P(E|\mathbf{N}_{\mathcal{S}} = \mathbf{n}_{\mathcal{S}})$ when it is clear from the context.

Further, $C(\mathcal{R}|\mathcal{S})$ is actually a function of $\mathbf{n}_{\mathcal{R}}$ and $\mathbf{n}_{\mathcal{S}}$. For fixed $\mathbf{n}_{\mathcal{S}}$, define the set $B := \{\mathbf{n}_{\mathcal{R}} | C(\mathcal{R}|\mathcal{S}) \geq \alpha\}$. It can equivalently be described by

$$B = \{\mathbf{n}_{\mathcal{R}} | \log \frac{P(E|\mathbf{n}_{\mathcal{R}}, \mathbf{n}_{\mathcal{S}})}{P(E|\mathbf{n}_{\mathcal{S}})} \geq \alpha\} = \{\mathbf{n}_{\mathcal{R}} | P(E|\mathbf{n}_{\mathcal{R}}, \mathbf{n}_{\mathcal{S}}) \geq P(E|\mathbf{n}_{\mathcal{S}}) \cdot e^\alpha\}.$$

We thus have

$$P(E|\mathbf{n}_{\mathcal{R}} \in B, \mathbf{n}_{\mathcal{S}}) \geq P(E|\mathbf{n}_{\mathcal{S}}) \cdot e^\alpha.$$

Hence,

$$\frac{P(E, \mathbf{n}_{\mathcal{R}} \in B | \mathbf{n}_{\mathcal{S}})}{P(\mathbf{n}_{\mathcal{R}} \in B | \mathbf{n}_{\mathcal{S}})} \geq P(E|\mathbf{n}_{\mathcal{S}}) \cdot e^\alpha.$$

Using

$$P(E|\mathbf{n}_S) \geq P(E, \mathbf{n}_R \in B|\mathbf{n}_S)$$

we obtain

$$P(\mathbf{n}_R \in B|\mathbf{n}_S) \leq e^{-\alpha}.$$

D Visualising polytrees

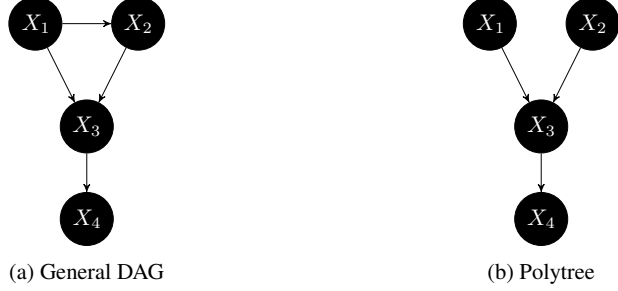


Figure 2: (a) A general DAG may contain undirected cycles, as between $X_1 - X_2 - X_3 - X_1$. (b) A polytree cannot contain undirected cycles so could not have an edge between X_1 and X_2 without first removing another.

E Is increase of scores along causal pathways rare?

Li et al. [46] describe a condition under which the z^2 -score of the effect X_j can be larger than the z^2 -score from the cause X_i in a DAG with linear structural equations. Their Theorem 2.1 states that this happens (in the limit of large perturbations, see the details further below) if and only if the variances σ_i^2, σ_j^2 of X_i, X_j satisfy

$$\sigma_j^2 < \alpha_{i \rightarrow j}^2 \sigma_i^2, \quad (32)$$

where $\alpha_{i \rightarrow j}^2$ denotes the structural coefficient for the effect from X_i to X_j (which consists of the sum over the product of all direct influences along the paths from i to j). Although our results refer to IT scores instead of z^2 -scores, it is at the same time an example for increasing IT scores since the z^2 -score can be turned into an IT score by a monotonic recalibration. It is therefore important that we understand this result if we are to understand if we expect SCORE ORDERING to be applicable in cases beyond polytrees. For an intuitive understanding we should first note that (32) describes the regime of *strong negative confounding*. This is because an unconfounded causal influence $X_j = \alpha_{i \rightarrow j} X_i + E_j$ with independent error term E_j results in $\sigma_j^2 = \alpha_{i \rightarrow j}^2 \sigma_i^2 + \text{Var}(E_j)$. We are therefore in the regime where the noise is so strongly negatively correlated with the cause that it decreases the variance instead of increasing it.

Let us now discuss whether this phenomenon is rare or not when X_i and X_j are variables in a larger network (note that the following working is strongly inspired by [46]). To this end let X_1, \dots, X_n be causally ordered with structural equations

$$\mathbf{X} = \mathbf{A}\mathbf{X} + \mathbf{N}, \quad (33)$$

where \mathbf{A} is a strictly lower triangular matrix and \mathbf{N} is the vector of independent noise variables. To simplify notation, let all variables have zero mean.

The covariance matrix $\Sigma_{\mathbf{X}\mathbf{X}}$ can be derived by well-known algebra:

$$\Sigma_{\mathbf{X}\mathbf{X}} = (\mathbf{I} - \mathbf{A})^{-1} \Sigma_{\mathbf{N}\mathbf{N}} (\mathbf{I} - \mathbf{A})^{-T},$$

where $\Sigma_{\mathbf{N}\mathbf{N}}$ denotes the covariance matrix of the noise, which is diagonal by assumption. Defining

$$\mathbf{L} := (\mathbf{I} - \mathbf{A})^{-1} \Sigma_{\mathbf{N}\mathbf{N}}^{1/2},$$

we obtain the unique Cholesky decomposition of $\Sigma_{\mathbf{X}\mathbf{X}}$ since $\Sigma_{\mathbf{X}\mathbf{X}} = \mathbf{L}\mathbf{L}^T$, where \mathbf{L} is lower diagonal (with non-zero diagonal by construction if we assume non-zero noise variance). The matrix \mathbf{L} has

interesting interpretations:

(i) L generates the observations from independent noise variables as sources, formally $\mathbf{X} = L\tilde{\mathbf{N}}$, where $\tilde{\mathbf{N}} := \Sigma_{\mathbf{NN}}^{-1/2}\mathbf{N}$ is the standardized version of the noise from the original structural equation (33).

(ii) The squared row sums of L thus show the variance of the observed variables, that is, $\sum_j L_{ij}^2 = \sigma_i^2$. Henceforth, we will assume it to be 1 without loss of generality.

(iii) The i th column of L describes how shifting the noise \tilde{n}_i to $\tilde{n}_i + 1$ changes the observations \mathbf{x} , that is, the observation $\mathbf{x} = L\tilde{\mathbf{n}}$ is changed to $\mathbf{x} + Le_i$, where e_i denotes the i th canonical basis vector.

Since all the X_j are standardized and centered, their z^2 score is simply their squared value x_j^2 . If we shift the noise \tilde{n}_i by a large value, the z^2 -scores x_i^2 of the root cause X_i versus the score x_j^2 of the effect X_j satisfies the ratio L_{ii}^2/L_{ij}^2 in the limit of infinitely strong perturbation. The question of whether scores of effects are typically smaller than scores of their root causes now boils down to the following question: Given a lower triangular matrix L whose row vectors have norm 1. Is L_{ij}^2 "typically" smaller than L_{ii}^2 ? We will discuss this question for the assumption that n is large and that $\rho := \min_i \{L_{ii}^2\}$ is much larger than $1/n$. This assumption simply formalizes the idea that no X_i is close to be a deterministic function of its ancestors. It may seem strong, but we could easily extend the below discussion to the case where only *most* of the variables satisfy this condition.⁹ We conclude:

$$\sum_{1 \leq i < j} L_{ij}^2 \leq 1 - \rho.$$

Hence, the number k of i with $i < j$ for which $L_{ii}^2 < L_{ij}^2$ is at most $(1 - \rho)/\rho$. Using $\rho \gg 1/n$ we conclude $k \ll n$. Thus, only for a small fraction k/n of root causes i , the score L_{ii}^2 of the root cause is smaller than the score L_{ij}^2 of the downstream effect j . In this sense, we may consider effects with larger score than the root cause as a rare phenomenon, even for *fixed* causal structural equations, not only as a statistical statement over multiple randomly generated structures.

F Experimental details and further experiments

F.1 Experimental details

To run Circa and Counterfactual, we used the implementations available in [27] using default parameters. We wrote our own implementation of Traversal (with anomaly score threshold of 3), and our own algorithms, SMOOTH TRAVERSAL AND SCORE ORDERING. The code for Cholesky is available at [46], however, there are in fact three different algorithms. For all experiments involving real-world data (see subsection F.6 below) we applied their "main" algorithm, using default parameters. For the synthetic experiments, we had to opt for the "highdim" version as the alternatives had already become prohibitively slow for the graph sizes considered. To generate an SCM for the experiments in Section. 4 (see Fig. 1), we first uniformly sample between 10 and 20 root nodes (20% to 40% of the total nodes of the graph) and uniformly assign to each either a standard Gaussian, uniform, or mixture of Gaussians as its noise distribution. As a second step, we recursively sample non-root nodes. Non-root nodes need not be sink nodes. The number of parent nodes that each non-root node is conditioned on is randomly chosen following a distribution that assigns a lower probability to a higher number of parents. In total, the causal graph is composed of 50 nodes. The parametric forms of the structural equations are randomly assigned to be either a simple feed-forward neural network with a probability of 0.8 (to account for non-linear models) and a linear model. The feed-forward neural network has three layers (input layer, hidden layer, and output layer) where the hidden layer has a number of nodes chosen randomly between 2 and 100. All the parameters of the neural network are sampled from a uniform distribution between -5 and 5. For the linear model, we sample the coefficients of the linear model from a uniform distribution between -1 and 1 and set the intercept to 0. In both cases, we use additive Gaussian noise as the relation between the noise and the variables.

⁹Note that *simulated* data can easily violate this assumption since many simulation schemes result in artifacts where variables that are late in the causal order can be almost perfectly reconstructed from others, a phenomenon called R^2 -sortability in [54]

To generate data for the non-anomalous regime, we sample the noise of each of the variables and propagate the noise forward using the previously sampled structural equations. As mentioned in the main text, to produce anomalous data, we choose a root cause at random from the list of all nodes and a target node from the set of its descendants (including itself). Then we sample the noise of all variables and modify the noise of the root cause by adding x -times standard deviations (of its marginal distribution), where $x \in \{2, 2.1, 2.2, \dots, 3\}$, and propagate the modified noise through the SCM to obtain a realisation of each variable. We repeat this process 100 times for each x value added to the standard deviation and consider the algorithm successful if its result coincides with the chosen root cause.

Computing Infrastructure. All the experiments were run on a MacBook Pro with 16GB of memory with an Apple M1 processor.

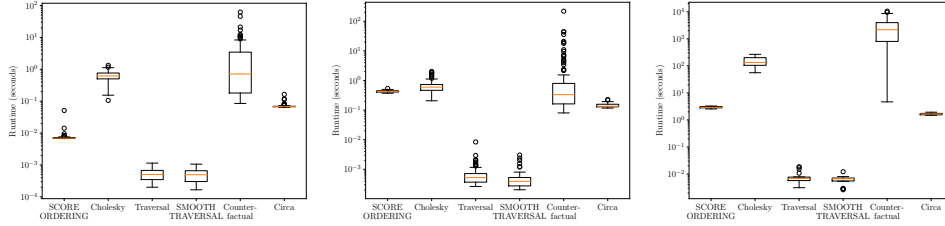


Figure 3: Runtimes of the algorithms for the experiment in Fig. 1 that is 50 nodes (left), an SCM with 100 nodes (center) and one with 1k nodes (right) (refer to the generation process above). The boxplots are produced using the default implementation in Matplotlib [55]. Note the log scale in the vertical axis.

F.2 Further experiments varying root cause anomaly strength

Methods which require multiple samples from the anomalous regime Most RCA approaches require multiple samples from the anomalous regime. For completeness, we extended the evaluation in Figure 1 to include two such methods: RCD [29] and ε -Diagnosis [11], as neither require the causal graph (see Figure 4). For both methods we made use of the implementations available in the PyRCA library [56] using default parameters. While all other methods were provided with only a single sample from the anomalous regime, both RCD and ε -Diagnosis were provided with 100, and so we believe that direct comparison is not appropriate. The experiment was otherwise identical to that described above in Appendix F. Nonetheless, we see that while RCD has very strong performance, with a top-1 recall close to one at all anomaly strengths, ε -Diagnosis is the opposite, with a recall close to zero for all anomaly strengths. It is worth noting that the original authors of the RCD method [10] show in their supplemental material that performance is very poor in the low-sample regime (<100 anomalous samples, see Figure 2 therein). We conclude that RCD could be preferable in scenarios where one has access to many samples in the anomalous period, but that in the low-sample regime SCORE ORDERING is preferable.

How do the algorithms perform when the structural causal model is linear? To test whether the performance of Circa and Cholesky in Fig. 1 was poor compared to the other algorithms due to the assumption of linearity, we reproduced the experiment as detailed in Section 4 and Appendix F above, with the only change being that now all structural equations were sampled to be linear. Indeed, Fig. 5 shows that Circa now becomes one of the best performing algorithms, alongside SMOOTH TRAVERSAL, Traversal, and Counterfactual. Similarly, although Cholesky remains below the top performers, its performance is notably improved at all anomaly strengths, even now outperforming SCORE ORDERING.

How do the algorithms perform when the causal graph is a polytree? In all synthetic experiments, we placed no restrictions on the structure of the causal graph other than that it be a DAG. Our main theorems are stated, however, for polytrees. We therefore repeated the experiments from Figure 1 with the single change constraining the simulated causal graphs to be polytrees (see Figure 6). Unsurprisingly, methods which do not make use of the causal graph have unchanged performance, and those that do are either unchanged or show improvement, including SMOOTH TRAVERSAL.

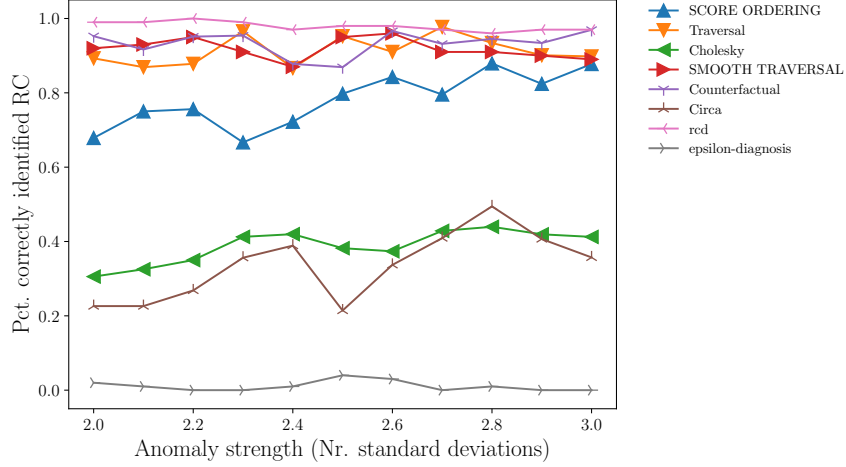


Figure 4: True positive rate for identifying the root cause against anomaly strength injected at the root cause, including RCD and ε -Diagnosis. Note that both RCD and ε -Diagnosis are given 100 samples from the anomalous period, while all other methods are given only one.

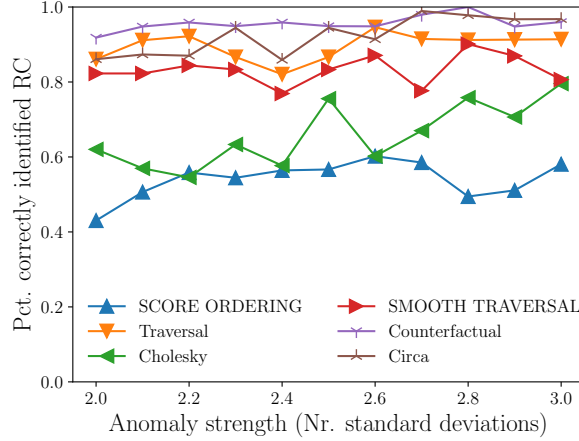


Figure 5: True positive rate for identifying the root cause against anomaly strength injected at the root cause, when all structural equations are linear.

Interestingly, Circa shows a marked improvement in performance despite not depending on the polytree assumption.

F.3 Further experiments by varying the number of nodes in the graph

We also run experiments by fixing the number of standard deviations added (x in Section. 4) to 3 and varying the number of nodes in the SCM. Here, we have chosen the numbers $\{20, 40, 60, 80, 100\}$. We observe in Fig. 7 that the performance for Traversal, SMOOTH TRAVERSAL, and Counterfactual does not change much for different graph sizes, whereas the performance of SCORE ORDERING, Cholesky and Circa decreases slightly for larger graph sizes.

F.4 How does performance change when the given causal graph is misspecified

To mimic the scenario where one has access to an imperfect causal graph, we investigate how robust the performance each of the algorithms (which require a causal graph) is to its misspecification (8). The data were generated as described at the beginning of Appendix F but with a fixed anomaly

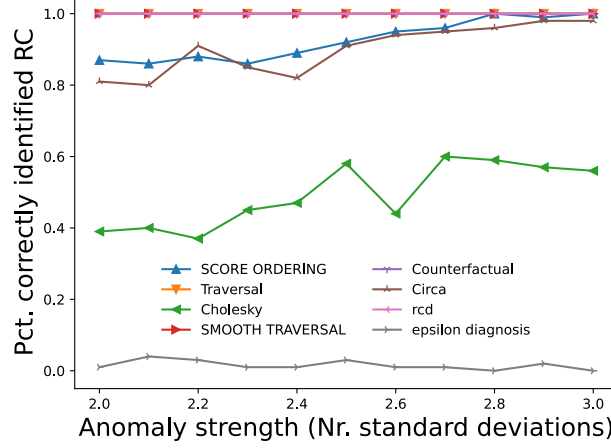


Figure 6: True positive rate for identifying the root cause against anomaly strength injected at the root cause, when all causal graphs are polytrees. Note that RCD and ϵ -Diagnosis are given 100 samples from the anomalous period, while all other algorithms are given only one.

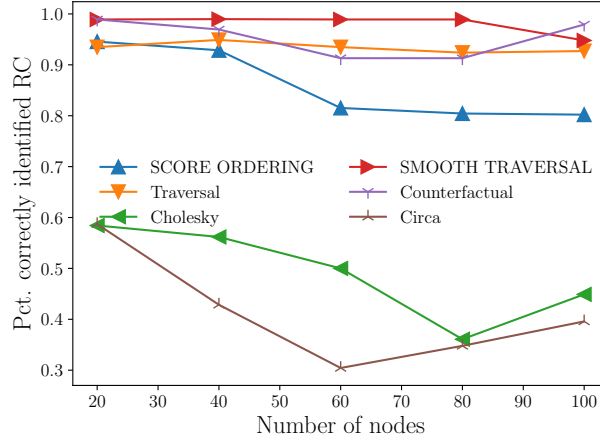


Figure 7: True positive rate of the compared algorithms in identifying the root cause with increasing numbers of nodes.

strength of $x = 3.0$, all structural equations linear, and the graph provided to each algorithm randomly altered to introduce mismatches with the true causal graph.

Starting from the sampled, true causal graph, random edge additions, removals and reversals were introduced according to a desired target Structural Hamming Distance [57, 58] from the true graph. To control for the density of the graph, an equal number of edges are randomly added as removed. In particular, for true causal graph $\mathcal{G} = (V, E)$, the total number of edges added and removed is sampled uniformly at random from $n_{\text{add/rem}} = [\text{SHD}(\mathcal{G}, \mathcal{G}_{\text{alt}}) - |E|, \text{SHD}(\mathcal{G}, \mathcal{G}_{\text{alt}})]$, with the number of edges flipped $n_{\text{flip}} = \text{SHD} - 2 \cdot n_{\text{add/rem}}$. Of the edges in \mathcal{G} , $n_{\text{add/rem}}$ are randomly selected for removal, and an equal number of ‘missing’ edges are selected for addition. Of the remaining ‘un-removed’ edges in \mathcal{G} , n_{flip} are reversed. If the resulting graph is not a DAG, the whole process is repeated until it is. The procedure for generating a randomly misspecified graph given a ground truth matches the “edge domain expert” procedure described in [59].

We chose to sample all linear structural equations as the relationship between top-1 recall and SHD was the same as when sampling non-linear structural equations for SMOOTH TRAVERSAL, Traversal and Counterfactual, but was more noteworthy for Circa. The reason being that the performance for

Circa is already so low for non-linear structural equations, that a considerable drop in performance was not easily perceptible.

We observe that all four evaluated algorithms show a considerable drop in performance as the SHD of the given graph from the true causal graph increased, with the most pronounced drop occurring for Circa. However, even when half of all edges are removed/flipped or added elsewhere in the graph ($\text{SHD}/|E| = 1$), SMOOTH TRAVERSAL, Traversal and Counterfactual still achieve a top-1 recall of approximately 0.6 to 0.7, showing that all three algorithms are relatively robust to graph misspecification.

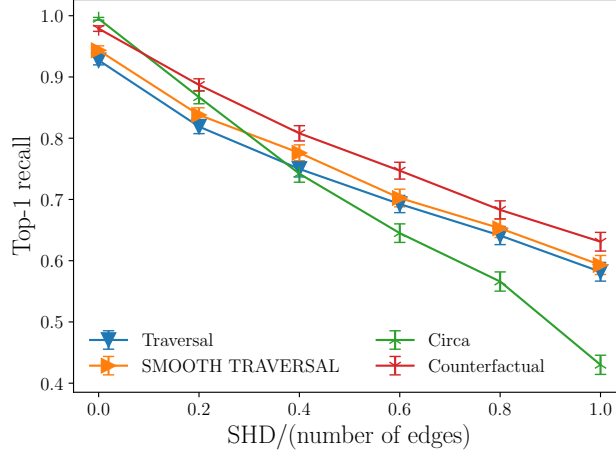


Figure 8: Top-1 recall for identifying the root cause using a graph with an increasing SHD (average per edge) from the true causal graph.

F.5 How do the algorithms compare in terms of their practicality?

It is important to keep in mind when interpreting the performances of each of the algorithms their different input requirements. Counterfactual either needs to be provided the SCM or must attempt to learn it from data. Traversal, Circa and SMOOTH TRAVERSAL require the causal graph as input but with differing assumptions on its structure. Traversal makes no assumptions, Circa assumes the SCM is linear and attempts to learn it from data, given the graph, and SMOOTH TRAVERSAL has guarantees only for the case the graph is a polytree (though it may be applied even if this assumption is not met). Traversal, however, requires an additional tuning parameter: the anomaly score threshold, while SMOOTH TRAVERSAL does not have any such free parameter. Both SCORE ORDERING and Cholesky do not even require the causal graph, however, Cholesky assumes the underlying SCM is linear. In synthetic data experiments, SCORE ORDERING consistently outperformed Cholesky other than when the SCM was restricted to be linear (see Fig. 3 above). Cholesky also scales more poorly in terms of run-time (see Fig. 3) due to the need to estimate the covariance matrix. In comparison, SCORE ORDERING makes very weak demands on the input, is efficient, and consistently performs well across experiments.

F.6 PetShop dataset

Hardt et al. [27] introduce a dataset from cloud computing, specifically designed for evaluating root cause analyses in microservice-based applications. The dataset is collected from a microservice application and includes 68 injected performance issues, which increase latency and reduce availability throughout the system. Latency and availability are so-called ‘metrics’ of the microservice application, and measure the length of time it takes for a request to the service to be processed, and the fraction of the time a request returns an error, respectively. The dataset also encompasses three traffic scenarios: low, high, and temporal. These correspond to the amount and periodicity of user traffic to the application. This presents a challenging task due to high missingness, low sample sizes, and near-constant variables. Furthermore, the ground truth causal graph is only partially known, as we must treat the edge-inverted call graph as a proxy for the causal graph.

In addition to the approaches evaluated by Hardt et al. (see [27] for more details), which we have reproduced below, we evaluated our algorithms in both top-1 recall (Table. 1) and top-3 recall (Table. 2). In addition, we included, as an additional baseline, the Traversal algorithm as described in Section 4. This is added as although the evaluation already includes a ‘Traversal’ algorithm implemented by Hardt et al., here called ‘petshop Traversal’, their algorithm does not match the standard approach as described in [9, 24, 25]. Instead, ‘petshop Traversal’ has been designed for use on the PetShop dataset, takes into account particular aspects of that data, and contrary to standard traversal algorithms, scores potential root causes according to anomalousness of whole paths of nodes from root cause to target node. ‘Petshop Traversal’ should not, therefore, be treated as an ‘out-of-the-box’ baseline.

We observe that SMOOTH TRAVERSAL and SCORE ORDERING perform very well in top-1 and top-3 recall, often being the best performing algorithms, and clearly outperforming simple Traversal and Cholesky. However, while SMOOTH TRAVERSAL always matches or exceeds the performance of simple Traversal in top-1 recall, on latency issues it nonetheless tends to perform relatively poorly compared to SCORE ORDERING.

Table 1: Top-1* recall, with ties. *Multiple anomalies can receive the same, highest, anomaly score. The root cause being in the top-1 means that it is among the variables receiving the highest score.

traffic scenario	metric	graph given			graph not given			this paper			
		petshop Traversal	Circa	counter-factual	ϵ -diagnosis	rcd	correlation	Traversal	Cholesky	SCORE ORDERING	SMOOTH TRAVERSAL
low	latency	0.57	0.36	0.36	0.00	0.07	0.43	0.14	0.29	1.00	0.14
low	availability	0.50	0.42	0.00	0.00	0.58	0.75	0.42	0.00	0.75	0.75
high	latency	0.57	0.50	0.57	0.00	0.00	0.64	0.14	0.14	1.00	0.14
high	availability	0.33	0.00	0.00	0.00	0.00	0.83	0.33	0.00	1.00	1.00
temporal	latency	1.00	0.75	0.38	0.12	0.25	0.62	0.25	0.50	0.75	0.25
temporal	availability	0.38	0.38	0.00	0.00	0.50	0.62	0.25	0.00	1.00	1.00

Table 2: Top-3* recall, with ties. *Multiple anomalies can receive the same, highest, anomaly score. The root cause being in the top-3 means that it is among the variables ranked in the top three including all variables tied at the top.

traffic scenario	metric	graph given			graph not given			this paper			
		petshop Traversal	Circa	counter-factual	ϵ -diagnosis	rcd	correlation	Traversal	Cholesky	SCORE ORDERING	SMOOTH TRAVERSAL
low	latency	0.57	0.86	0.71	0.00	0.21	0.57	0.14	0.57	1.00	0.86
low	availability	1.00	1.00	0.42	0.00	0.75	0.92	0.50	0.00	1.00	1.00
high	latency	0.79	1.00	0.86	0.00	0.07	0.79	0.14	0.57	1.00	0.93
high	availability	1.00	0.00	0.00	0.33	0.00	0.92	0.33	0.00	1.00	1.00
temporal	latency	1.00	1.00	0.50	0.12	0.75	0.75	0.25	0.63	1.00	0.75
temporal	availability	1.00	1.00	0.25	1.00	0.12	0.75	0.25	0.00	1.00	1.00

Note, however, the caveat that owing to the small sample size in the normal period in the PetShop dataset, it is frequently the case that multiple anomalies receive the same highest IT anomaly score (see Table 3 for numbers and proportions of nodes with tied scores in each traffic scenario). In the most severe cases there can be as many as 20 nodes tied with the maximum score (in the high traffic scenario), but it is typically fewer than 10. The multiplicity of highest scores is due to the fact that for the issue at hand all these metrics were more extreme than ever sampled in the normal period. In this case, the root cause being in the top-1 means that it is among the variables receiving the highest score. This problem should become less severe for datasets where anomaly scorers are trained on a longer history. Since root cause analysis without any graphical information is a notoriously hard problem, returning short lists of potential root causes, even if they contain 10 or more candidates, can still result in a drastic reduction of the search space, and so is of vital use in real applications.

Table 3: Average number and proportion of nodes in each traffic scenario and metric with a tied marginal anomaly score

Traffic scenario	Metric	Average number	Average proportion
high traffic	Latency	18.26	41.50%
high traffic	Availability	16.46	37.41%
low traffic	Latency	9.96	23.72%
low traffic	Availability	6.54	15.57%
temporal traffic1	Latency	5.75	14.02%
temporal traffic1	Availability	7.00	17.07%
temporal traffic2	Latency	5.75	14.02%
temporal traffic2	Availability	7.12	17.38%

We, nonetheless, also report the top-1 (Table. 4) and top-3 (Table. 5) recall after randomly selecting among the nodes tied with the highest anomaly score. As expected, we observe an approx. 10x decrease in top-1 recall for Traversal, SCORE ORDERING and SMOOTH TRAVERSAL, corresponding to selecting at random from among the approx. 10 candidate root causes with tied highest scores. Similar decreases are present in top-3 recalls for Traversal and SCORE ORDERING whenever the true root cause is among those with the tied highest score. Note, however, that SMOOTH TRAVERSAL generally still has high top-3 recall even after random selection of the top-1 node, outperforming both Traversal, SCORE ORDERING and Cholesky in all but one evaluation. Note that the performance of Cholesky does not considerably drop after random selection of ties, as variables rarely receive the same score.

Table 4: Top-1 recall, with random selection among ties.

		this paper			
traffic scenario	metric	Traversal	Cholesky	SCORE ORDERING	SMOOTH TRAVERSAL
low	latency	0.02	0.29	0.10	0.02
low	availability	0.06	0.00	0.10	0.10
high	latency	0.01	0.14	0.06	0.01
high	availability	0.03	0.00	0.06	0.09
temporal	latency	0.04	0.50	0.10	0.04
temporal	availability	0.04	0.00	0.13	0.14

Table 5: Top-3 recall, with random selection among ties

		this paper			
traffic scenario	metric	Traversal	Cholesky	SCORE ORDERING	SMOOTH TRAVERSAL
low	latency	0.02	0.57	0.10	0.73
low	availability	0.10	0.00	0.31	0.31
high	latency	0.01	0.50	0.06	0.80
high	availability	0.03	0.00	0.06	0.09
temporal	latency	0.04	0.63	0.18	0.54
temporal	availability	0.04	0.00	0.13	0.14

F.7 Sock-shop dataset

Similar to the PetShop dataset described above, Pham et al. [26] introduces another dataset from cloud computing for evaluating root cause analyses. The dataset includes 125 injected performance issues encompassing five types of common faults: CPU hog - ‘cpu’, memory leak - ‘mem’, disk IO stress - ‘disk’, network delay - ‘delay’, and packet loss - ‘loss’. We again used the edge-inverted call graph for the Sock-shop application as a proxy for the causal graph. We evaluated the performance

of our algorithms in top-1 (Table 6) and top-3 (Table 7) recall. It is important to note that while for most algorithms we provided only a single anomalous sample (the first sample from the anomalous period) for RCD and ϵ -Diagnosis we provided all 361 anomaly samples and so direct comparison is not recommended: both algorithms were included rather for completeness of evaluation.

We observe that on three out of five issue types (delay, disk and loss), SCORE ORDERING is the top performing approach for top-1 and top-3 recall. In each instance, SMOOTH TRAVERSAL also performed competitively. For issue types cpu and mem, however, both SCORE ORDERING and SMOOTH TRAVERSAL performed poorly, although performance was comparable among all algorithms which were provided with only a single anomalous sample. Given both RCD and ϵ -Diagnosis performed considerably better for both issues, this would suggest that the effect of the fault was not easily discernible from only a single sample in the anomaly period.

As for PetShop, we also report the top-1 (Table 8) and top-3 (Table 9) recall after randomly selecting among nodes with tied ranks. Here we observe that although the performance of SMOOTH TRAVERSAL is competitive for delay, disk and loss issues, both SCORE ORDERING and SMOOTH TRAVERSAL are outperformed by Circa and/or Cholesky. As both algorithms assume linear causal models, and that both algorithms struggled in non-linear settings in our simulation studies (see Figure 1), this may suggest that the causal relationships in Sock-shop are well modelled by linear relationships. As before, for cpu and mem issues, while all methods which use only a single anomaly sample perform very poorly, RCD performs well.

Table 6: Top-1* recall, with ties. *Multiple anomalies can receive the same, highest, anomaly score. The root cause being in the top-1 means that it is among the variables receiving the highest score. [†]These methods were provided with all samples from the anomaly period, whereas all else were given only one.

	cpu	delay	disk	loss	mem
SCORE ORDERING	0.08	0.80	0.84	0.72	0.08
SMOOTH TRAVERSAL	0.00	0.72	0.76	0.60	0.00
Cholesky	0.04	0.56	0.48	0.40	0.12
Circa	0.04	0.52	0.68	0.64	0.16
Counterfactual	0.04	0.28	0.00	0.12	0.04
Traversal	0.04	0.80	0.76	0.72	0.12
rcd [†]	0.88	0.52	0.52	0.44	0.36
epsilon diagnosis [†]	0.24	0.08	0.16	0.20	0.00

Table 7: Top-3* recall, with ties. *Multiple anomalies can receive the same, highest, anomaly score. The root cause being in the top-3 means that it is among the variables ranked in the top three, including tied ranks. [†]These methods were provided with all samples from the anomaly period, whereas all else were given only one.

	cpu	delay	disk	loss	mem
SCORE ORDERING	0.24	0.92	0.92	0.88	0.16
SMOOTH TRAVERSAL	0.12	0.80	0.84	0.72	0.12
Cholesky	0.16	0.88	0.88	0.84	0.24
Circa	0.28	0.72	0.80	0.80	0.28
Counterfactual	0.24	0.76	0.52	0.64	0.40
Traversal	0.04	0.80	0.76	0.72	0.12
rcd [†]	1.00	0.52	0.56	0.44	0.52
epsilon diagnosis [†]	0.56	0.52	0.28	0.48	0.00

Table 8: Top-1 recall, with random selection among ties. [†]These methods were provided with all samples from the anomaly period, whereas all else were given only one.

	cpu	delay	disk	loss	mem
SCORE ORDERING	0.08	0.24	0.29	0.26	0.04
SMOOTH TRAVERSAL	0.00	0.49	0.50	0.47	0.00
Cholesky	0.04	0.56	0.48	0.40	0.12
Circa	0.04	0.52	0.68	0.64	0.16
Counterfactual	0.04	0.28	0.00	0.12	0.00
Traversal	0.02	0.41	0.37	0.47	0.02
rcd [†]	0.88	0.56	0.48	0.40	0.36
epsilon diagnosis [†]	0.24	0.08	0.16	0.20	0.00

Table 9: Top-3 recall, with random selection among ties. [†]These methods were provided with all samples from the anomaly period, whereas all else were given only one.

	cpu	delay	disk	loss	mem
SCORE ORDERING	0.24	0.67	0.70	0.68	0.14
SMOOTH TRAVERSAL	0.12	0.76	0.83	0.72	0.04
Cholesky	0.16	0.88	0.92	0.84	0.24
Circa	0.28	0.72	0.80	0.80	0.28
Counterfactual	0.24	0.80	0.48	0.64	0.28
Traversal	0.04	0.79	0.75	0.72	0.07
rcd [†]	1.00	0.56	0.56	0.44	0.48
epsilon diagnosis [†]	0.56	0.52	0.28	0.48	0.00

F.8 ProRCA

To provide evaluations of the algorithms in a semi-synthetic setting, we used the ProRCA package, introduced in [48]. ProRCA consists of a hand-crafted model of a retail service, with a known causal graph, allowing synthesis of data from real-world business scenarios, into which different types of anomalies can be introduced.

There are five types of anomalies that can be injected in the process, affecting different variables. We synthesised data following the introduction of four of these anomaly types, as listed in the first column in Table 10, performing 100 replicates of each. We excluded the ‘COGs’ anomaly type, which introduces an anomaly at the variable ‘UNIT_COST’, as for a range of strengths of the anomaly, we were unable to produce changes in the data that were statistically distinct from those in the non-anomalous regime.

We removed from consideration variables in the graph that had plain text values, such as ‘PROMO_CODE’ as these are unsuitable for RCA, but doing so introduced no unmeasured confounding and so does not affect the analysis. For each anomaly, we considered ‘PROFIT’ to be the target variable. ‘PROFIT’ is not a leaf node in the causal graph and the length of the paths between the root cause and the target variable varies between two and three. We applied Circa, Counterfactual, Traversal, and SMOOTH TRAVERSAL algorithms for which we are able to provide (the marginalised) causal graph, and the Cholesky algorithm and SCORE ORDERING, which only requires data from the observational and the anomalous regime. Only one anomalous sample was provided in each case. The resulting Top-1 recall of the algorithms is reported in Table 10.

The results are mixed. SCORE ORDERING performs well for two types of anomaly. In both cases, the methods that require the graph also perform well. However, requiring the graph for the RCA algorithm might be constraining in most real-world scenarios. There are two types of anomaly where SCORE ORDERING does not perform well: In ‘ExcessiveDiscount’ only the Counterfactual method provides a Top-1 recall above 30%, even for those methods requiring the graph as input. In ‘ReturnSurge’, the Cholesky method performs well (66%) and those methods for which the graph

Table 10: Top-1 recall, with random selection among ties.

	graph given			graph not given	this paper	
Type of anomaly	Circa	Counterfactual	Traversal	Cholesky	SCORE ORDERING	SMOOTH TRAVERSAL
ExcessiveDiscount	0.04	0.89	0.30	0.00	0.04	0.04
FulfillmentSpike	0.96	0.98	0.94	0.74	0.94	0.28
ReturnSurge	1.00	0.88	0.97	0.66	0.00	0.18
ShippingDisruption	0.94	0.94	0.94	0.47	0.95	0.27

is given provide a recall above 85%. In all types of anomalies, SMOOTH TRAVERSAL has a performance below 30%.

References

- [1] S Wibisono, MT Anwar, Aji Supriyanto, and IHA Amin. Multivariate weather anomaly detection using dbscan clustering algorithm. In *Journal of Physics: Conference Series*, volume 1869, page 012077. IOP Publishing, 2021.
- [2] Eric V Strobl and Thomas A Lasko. Identifying patient-specific root causes with the heteroscedastic noise model. *Journal of Computational Science*, 72:102099, 2023.
- [3] Eric V Strobl. Counterfactual formulation of patient-specific root causes of disease. *Journal of Biomedical Informatics*, page 104585, 2024.
- [4] Gian Antonio Susto, Matteo Terzi, and Alessandro Beghi. Anomaly detection approaches for semiconductor manufacturing. *Procedia Manufacturing*, 11:2018–2024, 2017.
- [5] Jellis Vanhoeyveld, David Martens, and Bruno Peeters. Value-added tax fraud detection with scalable anomaly detection techniques. *Applied Soft Computing*, 86:105895, 2020.
- [6] Sanjiv Das, Richard Stanton, and Nancy Wallace. Algorithmic fairness. *Annual Review of Financial Economics*, 15:565–593, 2023.
- [7] Dongjie Wang, Zhengzhang Chen, Jingchao Ni, Liang Tong, Zheng Wang, Yanjie Fu, and Haifeng Chen. Hierarchical graph neural networks for causal discovery and root cause localization. *arXiv preprint arXiv:2302.01987*, 2023.
- [8] Cheng-Ming Lin, Ching Chang, Wei-Yao Wang, Kuang-Da Wang, and Wen-Chih Peng. Root cause analysis in microservice using neural granger causal discovery. *arXiv preprint arXiv:2402.01140*, 2024.
- [9] Dewei Liu, Chuan He, Xin Peng, Fan Lin, Chenxi Zhang, Shengfang Gong, Ziang Li, Jiayu Ou, and Zheshun Wu. Microhecl: high-efficient root cause localization in large-scale microservice systems. In *Proceedings of the 43rd International Conference on Software Engineering: Software Engineering in Practice*, ICSE-SEIP '21, page 338–347. IEEE Press, 2021.
- [10] Azam Ikram, Sarthak Chakraborty, Subrata Mitra, Shiv Saini, Saurabh Bagchi, and Murat Kocaoglu. Root cause analysis of failures in microservices through causal discovery. *Advances in Neural Information Processing Systems*, 35:31158–31170, 2022.
- [11] Huasong Shan, Yuan Chen, Haifeng Liu, Yunpeng Zhang, Xiao Xiao, Xiaofeng He, Min Li, and Wei Ding. ϵ -diagnosis: Unsupervised and real-time diagnosis of small-window long-tail latency in large-scale microservice platforms. In *The World Wide Web Conference*, pages 3215–3222, 2019.
- [12] Meng Ma, Jingmin Xu, Yuan Wang, Pengfei Chen, Zonghua Zhang, and Ping Wang. Automap: Diagnose your microservice-based web applications automatically. In *Proceedings of The Web Conference 2020*, pages 246–258, 2020.
- [13] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.
- [14] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.
- [15] Charu C Aggarwal. *An introduction to outlier analysis*. Springer, 2017.
- [16] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. A review on outlier/anomaly detection in time series data. *ACM Computing Surveys (CSUR)*, 54(3):1–33, 2021.
- [17] Leman Akoglu. Anomaly mining: Past, present and future. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1–2, 2021.
- [18] Edwin M Knorr and Raymond T Ng. Finding intensional knowledge of distance-based outliers. In *Vldb*, volume 99, pages 211–222, 1999.

- [19] Barbora Micenková, Raymond T Ng, Xuan-Hong Dang, and Ira Assent. Explaining outliers by subspace separability. In *2013 IEEE 13th international conference on data mining*, pages 518–527. IEEE, 2013.
- [20] Ninghao Liu, Donghwa Shin, and Xia Hu. Contextual outlier interpretation. *arXiv preprint arXiv:1711.10589*, 2017.
- [21] Meghanath Macha and Leman Akoglu. Explaining anomalies in groups with characterizing subspace rules. *Data Mining and Knowledge Discovery*, 32:1444–1480, 2018.
- [22] Nikhil Gupta, Dhivya Eswaran, Neil Shah, Leman Akoglu, and Christos Faloutsos. Beyond outlier detection: Lookout for pictorial explanation. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I* 18, pages 122–138. Springer, 2019.
- [23] Kailash Budhathoki, Lenon Minorics, Patrick Bloebaum, and Dominik Janzing. Causal structure-based root cause analysis of outliers. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2357–2369. PMLR, 17–23 Jul 2022.
- [24] Pengfei Chen, Yong Qi, Pengfei Zheng, and Di Hou. Causeinfer: Automatic and distributed performance diagnosis with hierarchical causality graph in large distributed systems. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 1887–1895, 2014.
- [25] JinJin Lin, Pengfei Chen, and Zibin Zheng. Microscope: Pinpoint performance issues with causal graphs in micro-service environments. In *International Conference on Service Oriented Computing*, 2018.
- [26] Luan Pham, Huong Ha, and Hongyu Zhang. Root cause analysis for microservice system based on causal inference: How far are we? In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering, ASE ’24*, page 706–715, New York, NY, USA, 2024. Association for Computing Machinery.
- [27] Michaela Hardt, William Roy Orchard, Patrick Blöbaum, Elke Kirschbaum, and Shiva Kasiswathan. The petshop dataset — finding causes of performance issues across microservices. In Francesco Locatello and Vanessa Didelez, editors, *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pages 957–978. PMLR, 01–03 Apr 2024.
- [28] Yu Gan, Mingyu Liang, Sundar Dev, David Lo, and Christina Delimitrou. Sage: practical and scalable ml-driven performance debugging in microservices. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS ’21*, page 135–151, New York, NY, USA, 2021. Association for Computing Machinery.
- [29] Azam Ikram. Sock-shop data. <https://github.com/azamikram/rcd/tree/master/sock-shop-data>, 2023.
- [30] Amin Jaber, Murat Kocaoglu, Karthikeyan Shanmugam, and Elias Bareinboim. Causal discovery from soft interventions with unknown targets: Characterization and learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9551–9561. Curran Associates, Inc., 2020.
- [31] Nicola Gnecco, Nicolai Meinshausen, Jonas Peters, and Sebastian Engelke. Causal discovery in heavy-tailed models. *The Annals of Statistics*, 49(3):1755 – 1778, 2021.
- [32] Carlos Améndola, Benjamin Hollering, Seth Sullivant, and Ngoc Tran. Markov equivalence of max-linear Bayesian networks. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1746–1755. PMLR, 27–30 Jul 2021.
- [33] J. Pearl. *Causality*. Cambridge University Press, 2000.

- [34] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: Identification and confidence intervals. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(5):947–1012, 10 2016.
- [35] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.
- [36] Akash Maharaj, Ritwik Sinha, David Arbour, Ian Waudby-Smith, Simon Z. Liu, Moumita Sinha, Raghavendra Addanki, Aaditya Ramdas, Manas Garg, and Viswanathan Swaminathan. Anytime-valid confidence sequences in an enterprise a/b testing platform. In *Companion Proceedings of the ACM Web Conference 2023*, WWW ’23 Companion, page 396–400, New York, NY, USA, 2023. Association for Computing Machinery.
- [37] Stephen Bates, Emmanuel J. Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for outliers with conformal p-values. *The Annals of Statistics*, 2021.
- [38] Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008.
- [39] Jonas Peters, Joris M. Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 15(58):2009–2053, 2014.
- [40] Aram Ebtekar, Yuhao Wang, and Dominik Janzing. Toward universal laws of outlier propagation. In *Proceedings of the Forty-First Conference on Uncertainty in Artificial Intelligence*, UAI ’25. JMLR.org, 2025.
- [41] R. A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, 1925.
- [42] L. H. C. Tippett. *The Methods of Statistics: An Introduction Mainly for Workers in the Biological Sciences*. Williams and Norgate Ltd., London, UK, 1931.
- [43] Daniele Tramontano, Anthea Monod, and Mathias Drton. Learning linear non-Gaussian polytree models. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 1960–1969. PMLR, 01–05 Aug 2022.
- [44] Martin E. Jakobsen, Rajen D. Shah, Peter Bühlmann, and Jonas Peters. Structure learning for directed trees. *Journal of Machine Learning Research*, 23(159):1–97, 2022.
- [45] Davin Choo, Joy Qiping Yang, Arnab Bhattacharyya, and Clément L. Canonne. Learning bounded-degree polytrees with known skeleton. In Claire Vernade and Daniel Hsu, editors, *Proceedings of The 35th International Conference on Algorithmic Learning Theory*, volume 237 of *Proceedings of Machine Learning Research*, pages 402–443. PMLR, 25–28 Feb 2024.
- [46] Jinzhou Li, Benjamin B. Chu, Ines F. Scheller, Julien Gagneur, and Marloes H. Maathuis. Root cause discovery via permutations and cholesky decomposition, 2024.
- [47] Mingjie Li, Zeyan Li, Kanglin Yin, Xiaohui Nie, Wenchi Zhang, Kaixin Sui, and Dan Pei. Causal inference-based root cause analysis for online service systems with intervention recognition. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3230–3240, 2022.
- [48] Ahmed Dawoud and Shravan Talupula. Prorca: A causal python package for actionable root cause analysis in real-world business scenarios. *arXiv preprint arXiv:2503.01475*, 2025.
- [49] Kun Zhang, Zhikun Wang, Jiji Zhang, and Bernhard Schölkopf. On estimation of functional causal models: general results and application to the post-nonlinear causal model. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(2):1–22, 2015.

- [50] Julius Von Kügelgen, Abdirisak Mohamed, and Sander Beckers. Backtracking counterfactuals. In Mihaela van der Schaar, Cheng Zhang, and Dominik Janzing, editors, *Proceedings of the Second Conference on Causal Learning and Reasoning*, volume 213 of *Proceedings of Machine Learning Research*, pages 177–196. PMLR, 11–14 Apr 2023.
- [51] A. Balke and J. Pearl. Counterfactual probabilities: Computational methods, bounds, and applications. In R. Lopez D. Mantaras and D. Poole, editors, *Uncertainty in Artificial Intelligence*, volume 10. Morgan Kaufmann, San Mateo, 1994.
- [52] Dominik Janzing, Patrick Blöbaum, Atalanti A Mastakouri, Philipp M Faller, Lenon Minorics, and Kailash Budhathoki. Quantifying intrinsic causal contributions via structure preserving interventions. In *International Conference on Artificial Intelligence and Statistics*, pages 2188–2196. PMLR, 2024.
- [53] Oliver Ibe. *Markov Processes for Stochastic Modelling*. Elsevier, 2nd edition edition, 2013.
- [54] Alexander G. Reisach, Myriam Tami, Christof Seiler, Antoine Chambaz, and Sebastian Weichwald. A scale-invariant sorting criterion to find a causal order in additive noise models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS ’23, Red Hook, NY, USA, 2023. Curran Associates Inc.
- [55] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
- [56] Chenghao Liu, Wenzhuo Yang, Himanshu Mittal, Manpreet Singh, Doyen Sahoo, and Steven C. H. Hoi. Pyrca: A library for metric-based root cause analysis, 2023.
- [57] Silvia Acid and Luis M. de Campos. Searching for bayesian network structures in the space of restricted acyclic partially directed graphs. *J. Artif. Int. Res.*, 18(1):445–490, May 2003.
- [58] Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, Oct 2006.
- [59] Elias Eulig, Atalanti A. Mastakouri, Patrick Blöbaum, Michaela Hardt, and Dominik Janzing. Toward falsifying causal graphs using a permutation-based test. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI’25/IAAI’25/EAAI’25. AAAI Press, 2025.