

Daten-effiziente Aufklärung von kausalen Zusammenhängen
in technischen Systemen durch Aktive Aufklärung
und die Verwendung von Vorwissen

Zur Erlangung des akademischen Grades einer

Doktorin der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Josephine Rehak

Tag der mündlichen Prüfung: 16.12.2024

1. Referent/Referentin: Prof. Dr.-Ing. habil. Jürgen Beyerer
2. Referent/Referentin: Prof. Dr. rer. nat. Oliver Niggemann

Abstract

The thesis contributes to the computer-driven learning of causal relations and their use. Causal relations are known to be omnipresent and naturally comprehensible for humans, while also being relevant for every science. They can be represented as causal graphs with nodes representing the events or states, while directed edges represent the presence of causal relations between them.

We develop a novel kind of intervention to learn causal relations in experimental setups. This novel approach makes use of the natural spread of causal information in the underlying causal graph to draw conclusions about its structure. We inject a low-invasive, but identifiable signal in a chosen node and try to rediscover it in the other variables to gain clues about their relations. We demonstrate their use on several generated timeseries datasets.

This thesis also described new methods for the efficient representations of graph sets, which equally match the set of learned causal relations. For this task, we employ for example Binary Decision Diagrams and adjacency matrices based on ternary logic. Additionally, we propose an adapted version of the PC algorithm, which can use the novel representations, while it also may exceed the original version in time and the number of independence tests.

We perform an investigation on existing metrics, which evaluate learned causal graphs using ground truth. First, we introduce our own metric, which performs on a variety of different causal graph types and can use not only Directed Acyclic Graphs as ground truth, but also their superclass of Maximal Ancestral Graphs. Then, several criteria are introduced to evaluate the metrics' capabilities and the metrics are inspected for them.

Finally, we propose a new approach on how to use causal graphs to draw conclusions about potential root causes for discovered anomalies. We also propose an extension, which employs causal concepts of time to improve its recommendations of root causes. For demonstration, both approaches are applied on a robotic gripping process.

This work was partially funded by DFG FOR 5339 and by the subsidy fund 'Wertstromkinematik' of the KIT.

Kurzfassung

Diese Dissertation leistet einen Beitrag zum computergestützten Lernen von Kausalbeziehungen und deren Nutzung. Kausalbeziehungen sind bekanntlich allgegenwärtig und für den Menschen natürlich verständlich, aber auch für jede Wissenschaft relevant. Sie können als kausale Graphen dargestellt werden, wobei die Knoten die Ereignisse oder Zustände repräsentieren, während gerichtete Kanten kausale Beziehungen zwischen ihnen darstellen.

Wir entwickeln eine neue Art von Intervention, um kausale Beziehungen in Versuchsanordnungen zu lernen. Dieser neuartige Ansatz nutzt die natürliche Ausbreitung kausaler Informationen im zugrundeliegenden kausalen Graphen, um Rückschlüsse auf dessen Struktur zu ziehen. Wir injizieren ein minimalinvasives, aber identifizierbares Signal in einen ausgewählten Knoten und versuchen, es in den anderen Knoten wiederzufinden, um Hinweise auf ihre kausale Beziehungen zu erhalten. Wir demonstrieren ihre Anwendung an mehreren generierten Zeitreihendatensätzen.

In dieser Arbeit wurden auch neue Methoden zur effizienten Darstellung von Graphenmengen beschrieben, die ebenfalls zu den identifizierten Kausalbeziehungen passen. Für diese Aufgabe haben wir z.B. Binäre Entscheidungsdiagramme und Adjazenzmatrizen basierend auf ternärer Logik verwendet. Zusätzlich schlagen wir eine angepasste Version des PC-Algorithmus vor, die in der Lage ist, die neuartigen Repräsentationen zu verwenden, während sie gleichzeitig die ursprüngliche Version in Bezug auf Zeit und Anzahl der Unabhängigkeitstests übertrifft.

Wir führen eine Untersuchung bestehender Metriken durch, die gelernte kausale Graphen anhand von Grundwahrheiten bewerten. Zunächst stellen wir unsere eigene Metrik vor, die für eine Vielzahl verschiedener kausaler Graphen geeignet ist und nicht nur gerichtete azyklische Graphen, sondern auch deren Oberklasse, die Maximal Ancestral Graphen, als Grundwahrheit verwenden kann. Anschließend werden verschiedene Kriterien zur Bewertung der Fähigkeiten der Metrik eingeführt und für die untersuchten Metriken ausgearbeitet.

Schließlich schlagen wir einen neuen Ansatz zur Verwendung von kausalen Graphen vor, um Rückschlüsse auf mögliche Ursachen für entdeckte Anomalien zu ziehen. Wir stellen auch eine Erweiterung vor, die in der Lage ist, zeitliche Regel der Kausalität zu betrachten, um das Schließen auf mögliche Fehlerursachen noch zu verbessern. Eine beispielhafte Anwendung wird an einem Roboter-Greifprozess durchgeführt.

Diese Arbeit wurde teilweise finanziert durch DFG FOR 5339 und durch die Förderungsfinanzierung 'Wertstromkinematik' des KIT.

Contents

Abstract	i
Kurzfassung	iii
Notation	ix
1 Introduction	1
1.1 Motivation	1
1.2 Contributions and Thesis Organization	2
2 Fundamentals of Causal Research	5
2.1 Basic Definitions	5
2.2 Established Causal Graphs	6
2.2.1 Directed Acyclic Graphs	7
2.2.2 Complete Partial Directed Acyclic Graphs	7
2.2.3 Maximal Ancestral Graphs	8
2.2.4 Partial Ancestral Graphs	8
2.3 Causal Graphs for Timeseries Data	8
2.3.1 Timeseries Graphs	9
2.3.2 Summary Graphs	9
2.3.3 Functional Causal Models	10
2.4 Fundamentals of Structure Learning	10
2.4.1 Score-based Learning of Causal Structures	11
2.4.2 Constraint-based Learning of Causal Structures	11
2.4.3 Observation-driven Learning of Constraints	12
2.4.4 Experiment-driven Learning of Constraints	13
2.4.5 Constraint-based Learning using the PC Algorithm	15
2.5 Evaluation of Learned Causal Graphs	16
3 Signal-based Interventions to Gain Structure Constraints	19
3.1 Related Work	19
3.2 Background	20
3.2.1 Discrete Fourier Transform	20
3.2.2 Mueen's Algorithm for Similarity Search	21
3.2.3 Causal Timeseries Dataset Generator	21
3.3 Causal Constraints from Signal Injections	21
3.3.1 Spread of Signals in Causal Graphs	22

3.3.2	Learning Structure Constraints from Signal Injections	25
3.3.3	Learning Temporal Order Constraints from Signal Injections Causing Unique Signal Recoveries	26
3.3.4	Learning Temporal Order Constraints from Signal Injections Causing Recurrent Signal Recoveries	29
3.4	Approaches for Signal Injection and Recovery	31
3.4.1	Waveform Modulation-based Injection and Recovery	31
3.4.2	Frequency Modulation-based Injection and Recovery	31
3.5	Application Example of Signal Injections	32
3.5.1	Setup of the Test Environment and Measurements	32
3.5.2	Setup of the Waveform Modulation Approach	33
3.5.3	Setup of the Frequency Modulation Approach	34
3.5.4	Results	35
3.6	Discussion	35
3.7	Conclusion and Outlook	36
4	Efficient Representations of Causal Graph Equivalence Classes	37
4.1	Related Work	38
4.1.1	Constraints in Current Causal Structure Learning	38
4.1.2	Constraint-Solving Methods	38
4.2	Background	39
4.2.1	Ternary Kleene Logic	39
4.2.2	Binary Decision Diagrams	39
4.3	Adapted Matrices for Equivalence Class Representation	40
4.3.1	Ternary Adjacency Matrices	40
4.3.2	Edge Probability Matrices	42
4.4	BDDs for Equivalence Class Representation	43
4.5	BDD Computation Speedup by Rotation	44
4.6	Application on Example Equivalence Classes	45
4.6.1	Example Using Generic Data	45
4.6.2	Example Using Data from Karst Rock	46
4.7	Structure Learning with the Path-Constrained PC1	48
4.8	Speeding up PCPC1	49
4.9	Discussion	50
4.10	Conclusion and Outlook	51
5	Scoring Methods for the Evaluation of Discovered Causal Graphs	53
5.1	Related Work in Scoring Method Evaluations	53
5.2	Definition of the Normed Causal Edit Distance	54
5.2.1	Universal Causal Graph Representation	54
5.2.2	Calculating the Causal Edit Distance	55
5.2.3	Analysis of Parameter Choice	56
5.2.4	nCED with Maximum Ancestral Graphs as Ground Truth	58

5.3	Benchmark of Learning Scores	60
5.3.1	Investigation of Applicability	61
5.3.2	Experiment on Scoring Consistency and Sensitivity	62
5.3.3	Elaboration on Bounds and Normalization	63
5.3.4	Elaboration on G^* identity	64
5.4	Discussion	64
5.5	Conclusion and Outlook	65
6	Root Cause Analysis using Causal Graphs	67
6.1	Related Work	68
6.2	Basic Algorithm Setup	70
6.3	Algorithm Extension with Coarse Temporal Information	71
6.4	Application on Robotic Gripping Scenario	73
6.4.1	Application Environment	73
6.4.2	Causal Graph of the Gripping Process	74
6.4.3	Inspected Root Causes	75
6.4.4	Experimental Setup and Results	76
6.5	Discussion	79
6.6	Conclusion and Outlook	79
7	Conclusion of the Thesis	81
	Bibliography	83
	Own publications	93
	Patents	95
	Supervised student theses	97
	List of Figures	99
	List of Tables	101
	Acronyms	103

Notation

This chapter introduces the notation and symbols which are used in this thesis.

General notation

Scalars	Italic Roman lowercase letters	x, k
Sets	Calligraphic Roman uppercase letters	\mathcal{E}
Matrices	Bold Roman uppercase letters	M
Variables	Roman lowercase letters	e
Distributions	Uppercase letters	P

Numbers and indexing

\mathbb{N}	Natural numbers
\mathbb{N}_0	Natural numbers including zero (non-negative integers)
t	Discrete points in time
i, j	Indexing for objects, measurements and points

Chapter: Fundamentals of Causal Research

N	Number of variables
a, b, c, v	Causal variables
a_i	Sample of a at time point i if a is a timeseries
G	Learned causal graph
\mathcal{V}	Set of graph nodes representing variables
\mathcal{E}	Set of edges between variables
E	Edge adjacency matrix
$e_{a,b}$	Causal relation and edge from a to b
G^*	Ground true causal graph
\mathcal{V}^*	True set of variables representing graph nodes

\mathcal{E}^*	True set of edges between variables
\mathcal{U}	Equivalence class and set of potential causal graphs
\mathcal{C}	Set of Constraint on \mathcal{U}
\mathcal{A}	Set of all (binary) adjacency matrices
\mathcal{A}_{DAG}	Set of all (binary) adjacency matrices of DAGs
$\text{do}(a)$	Intervention on variable a
\mathcal{F}	Set of functions of a functional causal model
f_a	Function of variable a in a functional causal model
N_a	Additive noise of variable a in a functional causal model
$a \perp\!\!\!\perp b, b \perp\!\!\!\perp a$	Independence between variables a to b
$a \not\perp\!\!\!\perp b, b \not\perp\!\!\!\perp a$	Dependence between variables a and b
$a \perp\!\!\!\perp b c, b \perp\!\!\!\perp a c$	Conditional independence between variables a and b given c
$a \not\perp\!\!\!\perp b c, b \not\perp\!\!\!\perp a c$	Conditional dependence between variables a and b given c
$a \rightarrow b$	Known presence of an edge from a to b
$a \nrightarrow b$	Known absence of an edge from a to b
$a \rightarrow^k b$	Known presence of a path from a to b of length k
$a \nrightarrow^k b$	Known absence of a path from a to b of length k
$a \rightarrow^* b$	Known presence of a path from a to b of any length
$a \rightarrow^* a$	Known presence of a path from a to a of any length larger than zero
$a \nrightarrow^* b$	Known absence of a path from a to b of any length
$a \multimap b$	Presence or absence of the edge from a to b is not known
$a \multimap^* b$	Presence or absence of the path from a to b is not known
$[a \rightarrow b]$	Equivalence class constrained by the known presence of an edge from a to b
t_a	Temporal occurrence of variable a

Chapter: Signal-based Interventions to Gain Constraints on Causal Graphs

w	Introduced signal
n	Introduced signal
\mathcal{W}	Set of wavelets for experimentation
v_{inj}	Variable of wavelet injection
\mathcal{V}_{obs}	Set of variables inspected for signals
\mathcal{V}_{rec}	Set of variables with successful wavelet retrieval

$\text{do}(a, w)$	Intervention on variable a by injecting a signal w
$\text{rec}(a, w)$	Recovery of a signal w in variable a
$\text{do}(a, w, t_a)$	Intervention on variable a by injecting a signal w at time point t_a
$\text{rec}(a, w, t_a)$	Recovery of a signal w in variable a at time point t_a
t_a	Time point of recovery of a signal w in variable a
$t_{a,n}$	n th time point of recovery of a signal w in variable a
f_{sine}	Frequency of the periodic sine-based signal
f_{sample}	Sampling frequency
f	Frequency of the Fast Fourier Transform
n	Length of the observed timeseries

Chapter: Efficient Representations of Causal Graph Equivalence Classes

f	Boolean function
\top	Terminal node indicating valid assignments to f
\perp	Terminal node indicating invalid assignments to f
$0, -, 1$	False, undefined, true in ternary logic
$\mathcal{U}_{a,a}$	Equivalence class constrained by $a \rightarrow^* a$
$\mathcal{U}_{a,b}$	Equivalence class constrained by $a \rightarrow^* b$
f_{high}	Positive child node of the inspected node
f_{low}	Negative child node of the inspected node
\top, \perp	Positive and negative terminal node of a BDD
\mathcal{A}_2	Set of adjacency matrices in their binary representation
\mathcal{A}_3	Set of adjacency matrices in their ternary representation
$\mathcal{A}_{3,\mathcal{U}}$	Set of adjacency matrices in their ternary representation representing the equivalence class \mathcal{U}
E_2	Adjacency matrix in its binary representation
E_3	Adjacency matrix in its ternary representation
$\mathcal{L}_{\mathcal{U}}$	Edge probability matrix representing the equivalence class \mathcal{U}
$\mathbf{0}_N$	Null matrix of size N
#indep. tests	Number of required independence tests

Chapter: Scoring Methods for the Evaluation of Discovered Causal Graphs

$s(G_1, G_2)$	Scoring method comparing graph G_1 to graph G_2
TP	Number of edges present in G and G^*
FP	Number of edges present in G but absent in G^*
TN	Number of edges absent in G and G^*
FN	Number of edges absent in G but present in G^*
undirected FP	Number of undirected edges present in G but are absent direct edges in G^*
undirected FN	Number of undirected edges absent in G but are present direct edges in G^*
G_e	Graph without any edges
G_i	Graph where all edges of G^* were flipped
$G_{u,1}$	Graph where all directed edges of G^* were replaced by one undirected edge
$G_{u,2}$	Graph where all directed edges of G^* were replaced by two undirected edges

Chapter: Root Cause Analysis using Causal Graphs

r	Root cause variable
\mathcal{V}_A	Set of detected anomalies
\mathcal{V}_R	Set of root cause variables
$\mathcal{V}_{G,r}$	Set of observable variables of g depending on r
\mathcal{V}_L	Set of latent variables
\mathcal{V}_M	Set of observed variables
$JRCS(r)$	Jaccard Root Cause Score for root cause r
\mathcal{P}	Set of annotated process steps
$\mathcal{P}_{a \rightarrow b}$	Set of annotated process steps for edge $e_{a,b} = 1$
$\mathcal{P}_{r \rightarrow^* m}$	Set of process steps in which r may affect m for a path $r \rightarrow^* m$
$\mathcal{P}_{\{r \rightarrow^* m\}}$	Set of process steps in which r may affect m for all paths $\{r \rightarrow^* m\}_G$
$\{r \rightarrow^* m\}_G$	Set of paths from r to m in G
$\mathcal{T}_G(r)$	Set of tuples containing pairs of observable variables and individual process steps as inferred from G and r

\mathcal{T}_A	Set of tuples containing pairs of observable variables and individual process steps as provided from the anomaly detection step
-----------------	---

1 Introduction

1.1 Motivation

Animals and humans still outperform machine learning algorithms in various tasks. For one, machine learning algorithms have trouble generalizing and their performance may drop if the data-generating environment changes and a shift in the data distributions occurs [DAm22]. They mostly remain black boxes and lack explainability [Rib16] and may suffer from undetected spurious correlations in the data which negatively affect their deductions [Ye24]. Additionally, they still lack capabilities in drawing deductions about their environment by experimentation and from observations of temporal orders [Sch22].

In research, a domain specifically focuses on the principles of causality [Pea10, Pea09]. In its essence, machine learning and Artificial Intelligence (AI) are naturally entangled with causality and its consideration and integration into machine learning may lead to beneficial contributions to the described problems [Sch22].

This is because causality is an omnipresent concept more important than the concepts of space and time [Sch22] allowing the method transfer between domains. By nature, causal relations have the advantage of being explainable, as humans and animals naturally learn and use causal relations since infancy [Pen07]. Also, causal understanding empowers us to design the right interventions to learn about a systems setup and to achieve desired outcomes [Pea19].

Empowering artificial intelligence with skills in the identification, and use of causal relations will automatically contribute to the use of generalizing concepts, will add to naturally explainable AI, and will minimize the risk of drawing false conclusions based on correlations alone and may benefit various application domains at once.

The domain in causal research that has specialized in the identification of causal relations is called Causal Structure Learning (CSL) or Causal Discovery (CD). The learned causal structures are commonly represented in the form of causal graphs. The domain using causal information to conclude is called causal inference [Gly19]. While the learning of causal relations using experiments has become a standard [Fis36], the computer-driven learning of causal relations is still in its infancy and there is a lack of approaches in several aspects.

- I. The existing approaches for the intervention-driven learning of causal relations perform significant changes in the given system. They either perturb existing causal relations or alternate the existing probability distributions significantly to learn about causal relations [Ebe07, Ebe14].

- II. As nowadays knowledge about present or absent causal relations may come from experiments, observations and prior knowledge, varieties of specifically designed causal graphs and learning algorithms emerged to allow their combination [Mee95, Bro22, Aci03] without considering the use of more general principles for handling and representing such knowledge.
- III. The number of methods learning causal relations increases without their results being comparable [Zan22]. To allow their benchmarking, specifically tailored metrics are required to score their learned graphs precisely and efficiently.
- IV. Several machine learning algorithms target the detection of anomalies [Cha09] and root causes [Sol17], but few works consider the use of causal graphs to identify root causes, which are themselves unmeasurable, but have an identifiable effect on measurable variables.

1.2 Contributions and Thesis Organization

This thesis contributes to causal research in four different aspects, which are reflected in the organization of the chapters. First, a primer in Chapter 2 will introduce the basics of the domain of causal research. It is followed by the individual contributions.

In Chapter 3, we present a novel approach to learn about present or absent causal relations in a given system by performing targeted, low-invasive interventions to tackle aspect I. For this purpose, a signal is injected into a chosen variable and if we rediscover its remainder in the other variables, we may learn about their causal relation to the variable of injection.

- A new concept for learning causal relations using the low-invasive injection of signals is developed.
- We elaborate on how to gain knowledge about causal relations from signal recovery.
- Two techniques for the injection and recovery of signals are proposed and exercised.

In Chapter 4, we contribute to aspect II by providing general concepts for the efficient representations of potential causal graphs that correspond to the collected knowledge of present or absent causal relations. We also introduce an adapted version of the established PC algorithm to learn causal graphs using the novel representations.

- Several efficient representations of potential causal graph sets are proposed. This includes the use of ternary adjacency matrices, Binary Decision Diagrams and edge probability matrices.
- We extend the established PC algorithm for computationally and time-efficient causal discovery.

To contribute to aspect III, Chapter 5 covers an investigation of current metrics in the causal discovery domain. Specifically, we inspect metrics which use ground truth to evaluate the learned causal graphs. Additionally, we propose a new metric to allow the general benchmarking of causal discovery algorithms.

- A new metric called normalized Causal Edit Distance (nCED) is introduced.
- We demonstrate that the nCED cannot use DAGs as ground truth, as it is common, but also can use their superclass of Maximal Ancestral Graphs.
- A set of relevant criteria is proposed for causal structure metrics to proficiently evaluate discovered graphs.
- The established metrics and the nCED are inspected for the developed set of criteria.

In Chapter 6 we contribute to aspect IV by introducing two novel approaches for root cause analysis. We adapted causal graphs to enable the inference of potential root causes for a given set of detected anomalies.

- We developed an adaptation of causal graphs to support root cause analysis and coarse temporal annotations.
- We created a novel root cause inference approach, which scores potential root causes fitting to observed anomalies using causal graphs.
- We extended the root cause algorithm to draw causal-temporal valid deductions over potential root causes.

Finally, Chapter 7 summarizes our work and concludes the thesis.

2 Fundamentals of Causal Research

In Section 2.1, we describe the basic definitions of causation. Then in Section 2.2 and Section 2.3, we introduce the different definitions of causal graphs relevant to this thesis. Section 2.4 will introduce basic knowledge about the learning process of causal graph structures. Finally, Section 2.5 addresses basic knowledge about the evaluation of learned causal graphs.

2.1 Basic Definitions

A causal relation is a relationship between two or more variables representing events or states. Assume two variables a and b are given, and we know a causes b written as $a \rightarrow b$, identical to $b \leftarrow a$, then they are directly causally related. We call variables caused by other variables, such as b is caused by a , their descendants. In reverse, variables causing other variables, as is a causing b , are called their ancestors. For a relationship between variables to be causal, the following statements need to hold.

- **Dependence criterion**

If a causes b , then b depends on a for its value [Pea00].

- **Deletion criterion**

Since a causal related event depends on another event for its value, removing the cause a will also remove b .

- **Temporal delay criterion**

If a causal relation is present like $a \rightarrow b$, then a minimum delay time must be kept between the causing variable (a) and its dependent variable (b). This results from the laws of special relativity. These laws state that causal effect can only travel at the speed of light. Therefore, an event always sends a forward light cone to the future and a back light cone to the past in spacetime [Tho97, Win77]. Each ancestor or predecessor of the event must be in the appropriate light cone. A maximum delay time is not specified, as it has been shown that the effects of a causal relationship can be delayed.

- **Temporal order criterion**

The previous criterion also has further implications, as each causing variable can only influence variables in its future light cone. If for example $a \rightarrow b$ holds, then b cannot occur before a in time (assuming no other event besides a causes it). If we observe first a then b , b cannot have caused a as a is not in the future light cone of b . This implies the widespread knowledge that an effect cannot happen before its cause [Bee81]. In this work, refer to as the Temporal Law of Causation.

- **Manipulation criterion**

The relation $a \rightarrow b$ holds, when a manipulation (written as $\text{do}(a)$) for variable a exists, such that the value of the dependent variable b is affected [Rei44].

- **Correlation criterion**

A causal relation between variables implies a linear or non-linear correlation (often referred to as mutual information) of their values [Hla07]. Mind that in reverse, a correlation in their values does not necessarily imply a causal relation [And22].

- **Independence criterion**

The absence of a causal relation between variables is the independence of variables. This is denoted as $a \perp\!\!\!\perp b$, then neither a depend on b , nor does b depends on a for their respective values. Independence can be tested by computing if for their probability distributions holds $P(a)P(b) = P(a, b)$. If a relation between a and b is present $a \not\perp\!\!\!\perp b$, then the a test indicates their dependence $P(a)P(b) \neq P(a, b)$. Given a third variable c , a and b are conditionally independent given $a \perp\!\!\!\perp b \mid c$, if $P(a \mid b, c) = P(a \mid c)$ [Spi00]. To use probability distributions to detect causal relations, we assume the faithfulness assumption [Spi00, Pea00] to hold. It states that the observable probabilities are faithful to the underlying causal relations.

- **Causal sufficiency assumption**

To ensure a causal relation between a and b is actually present, we require their relation not to be spuriously implied because a hidden, third variable c is present, called confounder, that influences the inspected variables $a \leftarrow c \rightarrow b$ [Zha12a]. Such a hidden influence may result in wrong deductions about $a \rightarrow b$. Commonly, in CD the absence of latent confounders is assumed.

For each variable holds the Markov Condition. It states that any variable is independent of other variables given their causing variables (also called ancestors). For example, given $a \rightarrow b \rightarrow c$, this causes c to be independent of a given b also written as $a \perp\!\!\!\perp c \mid b$ identical to $c \perp\!\!\!\perp a \mid b$.

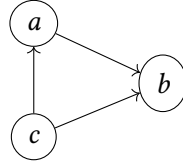
If a directed chain of at least two causal relations is present, as in $a \rightarrow b \rightarrow c$, also an indirect causal relation is present. In the example, a indirectly causes c . Such indirected causal relations can be represented as causal paths. We can write $a \rightarrow^k c$ to imply such a path of the length of k causal relations exists. Given $a \rightarrow b \rightarrow c$, then this can be abbreviated as $a \rightarrow^2 c$. By this definition, edges are paths in which $k = 1$ holds, thus $a \rightarrow b = a \rightarrow^1 b$. We use the notation $a \rightarrow^* c$ to indicate the existence of a path of causal relations of arbitrary length oriented in the direction of c . If we do not know if a certain causal relation between two variables exists, we write in this thesis $a \multimap b$. If we do not know if a causal relationship exists along a path, we write $a \multimap^* b$.

2.2 Established Causal Graphs

In literature, several kinds of causal graphs are established. In the following, we give a brief introduction to the types relevant to this thesis. Some contain only information about the bare

graph structure. As these graphs are the main outcome of causal discovery, our focus is set on these graph types. They all consist of a set of nodes representing variables \mathcal{V} and a set of edges \mathcal{E} representing direct causal relations between variables. \mathcal{E} can alternatively be represented as an edge matrix \mathbf{E} . On several occasions throughout this thesis, we will utilize principles, which use causal graphs as an underlying structure but contain additional features. We give a brief introduction to them as well.

2.2.1 Directed Acyclic Graphs



(a) Example DAG

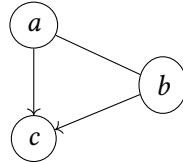
$$\mathbf{E} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

(b) Adjacency matrix

Figure 2.1: An example DAG and its adjacency matrix representation.

A common way to represent causal relations is in the form of Directed Acyclic Graphs (DAGs) [Pea00]. An example of a DAG and its adjacency matrix is depicted in Figure 2.1. In their graph representation, independencies and absent causal relations are implied by the absence of edges. Each DAG is uniquely represented by its adjacency matrix. In them, an edge between two variables $a \rightarrow b$ is present if $e_{a,b} = 1$ and absent $a \nrightarrow b$ if $e_{a,b} = 0$. These directed edges are forbidden to form cycles. Hence, any causal relation $a \rightarrow b$ always implies $b \nrightarrow a$ in DAGs.

2.2.2 Complete Partial Directed Acyclic Graphs



(a) Example CPDAG

$$\mathbf{E} = \begin{bmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

(b) Adjacency matrix

Figure 2.2: An example CPDAG and its adjacency matrix representation.

Complete Partially Directed Acyclic Graphs (CPDAGs), as shown in Figure 2.2, contain present edges, absent edges, also undirected edge pairs like $a - b$. In this case, it is a causal relation exists between a and b but its direction, if it is $e_{a,b} = 1$ or $e_{b,a} = 1$, is unknown. In the adjacency matrix, this is indicated by $e_{a,b} = -1$ and $e_{b,a} = -1$. CPDAGs represent multiple DAGs. To obtain them, given that no cycles are created, each undirected edge may be directed in either of its directions [Mee95]. Commonly, CPDAGs have directed edges, where colliding structures

such as $a \rightarrow c \leftarrow b$ are present. How such graphs are learned and how edges are oriented will be described in Section 2.4.5 in detail.

2.2.3 Maximal Ancestral Graphs

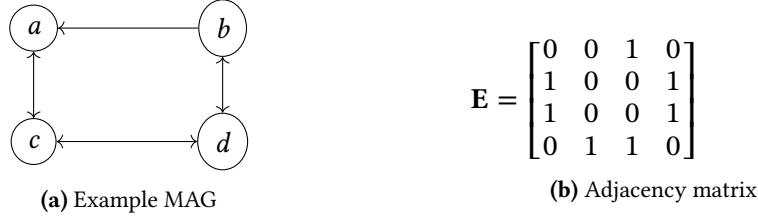


Figure 2.3: An example MAG and its adjacency matrix representation.

For the class of ancestral graphs, we consider Maximal Ancestral Graphs (MAGs). An example MAG is depicted in Figure 2.3. Additionally to DAGs, MAGs allow one-directed and bidirected edges, but forbid any cycles formed by a combination of bidirected and directed edges. In ancestral graphs, bidirected edges $a \leftrightarrow b$ imply the presence of a latent variable l with $l \rightarrow a$ and $l \rightarrow b$. DAGs are a subset of MAGs which do not contain any bidirected edges [Zha08a].

2.2.4 Partial Ancestral Graphs

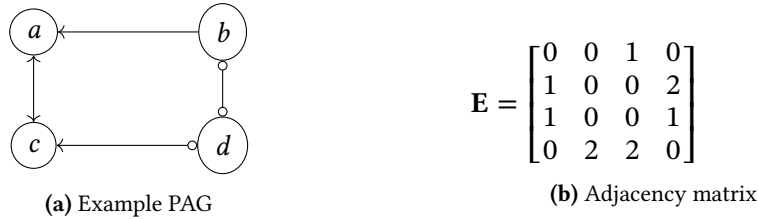


Figure 2.4: An example PAG and its adjacency matrix representation.

Partial Ancestral Graphs (PAGs) [Ric02, Ric96, Ric03], as shown in as Figure 2.4, may contain directed edges and absent edges. Additionally, they have edges to indicate missing information. This is represented by circular arrowheads in the graph and by assigning the value two to edges in the adjacency matrix. If these undirected edges are oriented, several MAGs are implied. Other than in CPDAGs, it is possible for opposing edges to be unoriented, but also single edges to be unoriented. Same as MAGs, they may also contain bidirected edges to indicate confounding variables.

2.3 Causal Graphs for Timeseries Data

Specific graphs have been developed to represent causal relations in timeseries data.

2.3.1 Timeseries Graphs

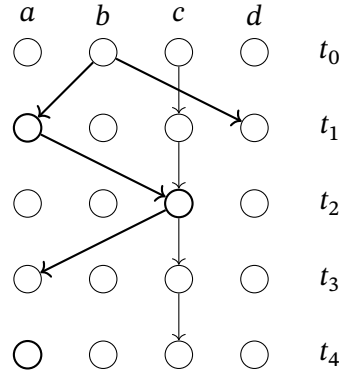


Figure 2.5: Depicted is a simple example of a time series graph. It contains discrete nodes for each variable and for each sampled timestep t_0, \dots, t_4 .

Timeseries graphs are causal graphs, which represent causal relations in multivariate timeseries data [Run19, Run12]. As shown in Figure 2.5, these timeseries graphs assume each variable in \mathcal{V} to be represented by a collection of nodes representing each sample of the corresponding timeseries. Directed edges indicate causal relations between the samples. Per the Temporal Law of causation, no node may cause another node in an earlier time step. The nodes of a variable may, but not necessarily need to, cause themselves over time leading to autoregressive behavior [Run19, Run12].

2.3.2 Summary Graphs

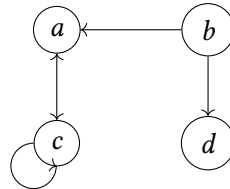


Figure 2.6: The corresponding summary graph for Figure 2.5.

So-called summary graphs [Ass22b, Run12] allow the compact depiction of time series graphs. We show in Figure 2.6, the summary graph for the timeseries graph in Figure 2.5. In them, each variable represents a timeseries. An edge $a \rightarrow b$ exists between the variables $a \in \mathcal{V}$ and $b \in \mathcal{V}$ if there exists any sample $i \in a$ causing an effect in a sample of $j \in b$ if $a \neq b$, such that i precedes or is equitemporal $i \leq j$ to b_j in time or if $a = b$ that i precedes j in time $i < j$. A

causal path $a \rightarrow^* c$ is present in the summary graph, if a sequence of directed edges from $i \in a$ to $j \in c$ for any i to j is present [Ass22a].

2.3.3 Functional Causal Models

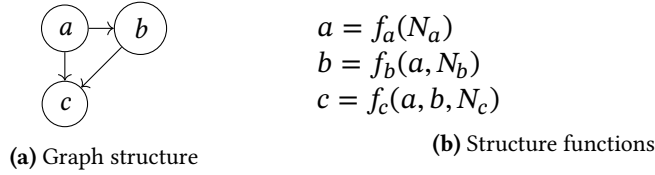


Figure 2.7: Minimal example of a functional causal model

Some kinds of graphs contain additional features besides the underlying causal structure. Functional Causal Models (FCMs) were first used by [Wri21]. They consist of a set of edges \mathcal{E} , a set of variables \mathcal{V} and a set of functions \mathcal{F} . Each variable $v \in \mathcal{V}$ has an assigned function $f_v \in \mathcal{F}$ that describes the causal relations between v 's parent variables, a noise factor N_v and v in detail. To ensure structure minimality, only non zero assignments are allowed to the variables in the functions. In Figure 2.7, we show a minimal example of a three variable FCM.

2.4 Fundamentals of Structure Learning

The overall goal of causal structure learning (CSL) is to identify a causal graph $G = (\mathcal{E}, \mathcal{V})$ in such a way that it is identical to the hidden ground truth $G^* = (\mathcal{E}^*, \mathcal{V}^*)$. Such a ground truth is assumed to be present in any system under inspection. Primarily, CSL concerns the correct discovery of causal relations. Commonly, the set of variables is considered fixed for a causal discovery problem and does not need to be discovered $\mathcal{V} = \mathcal{V}^*$.

Commonly, G^* is assumed to be a DAG, while in timeseries data, this graph is assumed to be a timeseries graph. G^* is assumed to be fully discovered if evidence for the absence or presence of all its edges are collected. The number of edges that require discovery in static graphs is $N(N-1)$ given $N = |\mathcal{V}|$. For timeseries graphs with M samples per variable, this requires the discovery of $MN(N-1)$ edges. As Table 2.1 shows, the number of potential graphs grows exponentially with the number of variables.

Table 2.1: In tabular data, the number of possible graphs grows exponentially with the number of variables N .

N	DAGs	MAGs
1	1	1
2	3	4
3	25	64
4	543	4,096
5	29,281	1,048,576
6	3,781,503	1,073,741,824
7	1,138,779,265	4,398,046,511,104
8	783,702,329,343	72,057,594,037,927,936

To reduce the number of potential graphs, we try to gather evidences on the true causal graph. The information may stem for example from probability distributions in the data, from prior knowledge or from information gathered by experiments. During the learning process, several graphs may be equally compliant with G^* for given information. As these sets of potential graphs fulfill the criteria of symmetry, reflexivity and transitivity regarding the provided knowledge, they form an equivalence class \mathcal{U} of graphs [Moo20, Squ23]. Several approaches came up in literature on how to learn causal structures and how to treat such equivalence classes. We feature relevant approaches in the following subsections.

2.4.1 Score-based Learning of Causal Structures

So-called score-based learning algorithms, try to optimize a score $S(G)$ by discovering the right G maximizing it. In an ideal scenario, the resulting graph of the search procedure is a single graph G . This G is assumed to be identical to the true causal graph. However, it occurs that several graphs are score equivalent given the probabilistic distributions in the data and thereby the graphs form an equivalence class [Squ23]. [Chi96] shows that popular scoring methods such as the Aikake information criterion [Aka98], the Bayesian information criterion, the Minimum Description Length criterion and the Bayesian Dirichlet equivalent criterion [Hec95] result in such score-equivalent graph sets.

A common approach is to apply heuristic search algorithms, which typically start with an empty graph and iteratively add or delete edges to increase the objective score of G to provided data and thereby to heuristically discover a DAG, assumed to be G^* [Moo20, Tsa06]. However, these algorithms are required to make a tradeoff between efficiency and algorithmic consistency, since this search for the highest-scoring DAG is shown to be NP-hard [Chi96].

Since this thesis primarily focuses on constraint-based learning, we recommend the work of Mooij et al. [Mag16, Moo20] for a detailed introduction.

2.4.2 Constraint-based Learning of Causal Structures

In the domain of constraint-based causal discovery, evidences about the absence or presence of causal relations are collected in the form of edge constraints. Graphs which do not comply with the constraints are pruned from the equivalence class. Constraints may stem from prior knowledge or from causal structure learning algorithms using observational or interventional data. If the set of unique collected constraints is complete and only one causal graph remains, this graph is assumed to be the ground truth $\mathcal{U} = \{G^*\}$.

If no constraints are given $\mathcal{C} = \emptyset$, the equivalence class \mathcal{U} is the set of all directed graphs. For simplicity, we represent the latter as a set of adjacency matrices $\mathcal{A} = \{0, 1\}^{N \times N}$. If we impose a present edge constraint like $a \rightarrow b$ on \mathcal{A} , all incompatible adjacency matrices are eliminated. We write such a small constrained equivalence class with angular brackets (e.g. $\mathcal{U} = [a \rightarrow b]$).

$$[a \rightarrow b] = \{\mathbf{E} \in \mathcal{A} : e_{a,b} = 1\} \quad (2.1)$$

The same holds for absent edge constraints.

$$[a \nrightarrow b] = \{E \in \mathcal{A} : e_{a,b} = 0\} \quad (2.2)$$

If two edge constraints are jointly imposed on equivalence classes, then holds the following:

$$[a \rightarrow b \wedge b \rightarrow c] = [a \rightarrow b] \cap [b \rightarrow c] \quad (2.3)$$

But if either of two constraints $a \rightarrow b$ or $b \rightarrow c$ is known, then holds:

$$[a \rightarrow b \vee b \rightarrow c] = [a \rightarrow b] \cup [b \rightarrow c] \quad (2.4)$$

If present or absent path constraints are provided, we may break them down into individual edge constrained equivalence classes by considering all potential combinations of edges for the given variables a, b, c .

$$[a \rightarrow^* c] = [(a \rightarrow c) \vee (a \rightarrow b \wedge b \rightarrow c) \vee (a \rightarrow c \wedge a \rightarrow b \wedge b \rightarrow c)]$$

The same holds for absent edge constraints.

$$[a \nrightarrow^* c] = [a \nrightarrow c \wedge (a \nrightarrow b \vee b \nrightarrow c)]$$

To restrict an equivalence class to contain only DAGs, it needs to hold $\mathcal{A}_{\text{DAG}} = [\forall v \in \mathcal{V}, v \nrightarrow^* v]$. For DAGs, the maximum number of present edge constraints is $\frac{N(N-1)}{2}$ given $N = |\mathcal{V}|$.

2.4.3 Observation-driven Learning of Constraints

Novel research has focused on the learning of causal relations from observational data alone [Pea00]. Several trends have developed to gain structure constraints from interventions.

Constraints from Prior Knowledge

The simplest way of gaining structure constraints is from human expertise. The provided constraint can for example include the orientation of edges to orient a CPDAG to a DAG [Mee95]. It may come in the form of a general concept of causal relations between variable types [Bro22]. For this purpose, Brouillard et al. grouped variables in types and provided typing assumptions, constraining the edges between variables of a specified type to orient CPDAGs. Other prior knowledge may include knowledge of the complete or partial temporal order of variables [ODo06]. According to the temporal law of causation, an effect cannot precede its cause, this is how constraints can be drawn for all non-conforming edges to be absent.

Constraints from Conditional Independence Tests

Naturally, this set of methods highly relies on the quality of the collected data. The most popular approach in causal research are Conditional Independence Test (CIT) which can be tested by the Fishers Z-test [Fis15] or kernel-based conditional independence tests [Zha12b]. They perform on

multivariate tabular data [Ver22] and as well on timeseries data [Run19] under the assumption of faithfulness. CITs allow the identification of a specific pattern in the data called a collider consisting of two directed edges. Consider this simplified construction block of a causal graph.

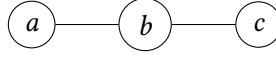


Figure 2.8: Simple building block of a causal graph with unoriented edges

The building block may take the form of the DAGs shown in Figure 2.9. While for chains, reverse

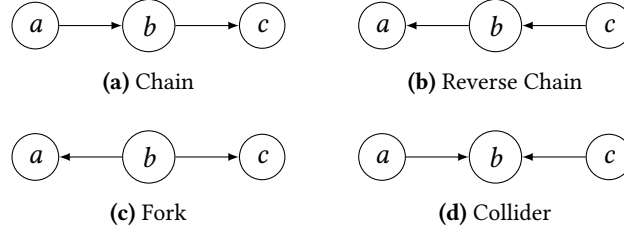


Figure 2.9: Conditional independence tests applied to given data can identify immoralities in the graph structure. chains and forks hold $a \perp\!\!\!\perp c \mid b$. This is not valid for colliders; for them holds $a \not\perp\!\!\!\perp c \mid b$ and thus their discovery is possible. Besides on such trivial examples, we can also test conditional independence $a \perp\!\!\!\perp b \mid \mathcal{P}$ using a set of variables \mathcal{P} .

Constraints from Additive Noise Models

Another approach to identifying the direction of a causal relation is by the use of Additive Noise Modelling (ANM) [Pet14]. Assuming the underlying causal graph is a functional causal model, and two variables a and b are given and for both an additive noise model generated the data given $a \rightarrow b$, then an additive noise model may be fitted for $a \rightarrow b$, but not in the direction of $b \rightarrow a$. This way, the direction between variables can be identified [Pet14]. Parida et al. [Par18] transferred this approach to multivariate applications.

2.4.4 Experiment-driven Learning of Constraints

Learning causal relations by experimentation is the gold standard in science to identify causal relations. Woodward et al. [Woo05] considered it a sufficient criterion to establish causal relationships. They stated that for two variables a and b holds $a \rightarrow b$ if there exists an intervention $\text{do}(a)$ for variable a , such that the value of b is affected on intervention on a . Throughout this work we will use the $\text{do}(a)$ -operator introduced by Pearl et al. [Pea12] to imply a variable a is physically manipulated without interfering on other variables.

Interventions in Causal Research

Eberhardt et al. [Ebe07] identified two main approaches in causal research on how to gain causal information by intervention in an experimental setup under the Markov assumption and the faithfulness assumption. Accordingly, they divided the existing literature into hard interventions and soft interventions.

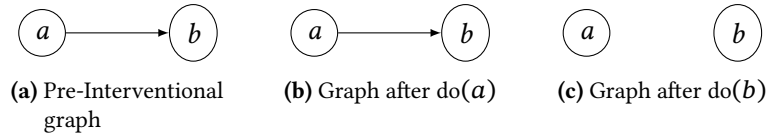


Figure 2.10: Shown is the effect of a hard intervention interventions on a given causal structure. A hard interventions eliminates all other influences on the variable of intervention.

Hard interventions [Pea09], also called structural, surgical or independent interventions [Kor04] take over control of a chosen variable. By doing so, other influences on the targeted variable are cut off and only we define its state or value.

For a demonstration of hard interventions consider Figure 2.10 and assume causal sufficiency, the absence of confounders. We try to identify the causal relations between a and b . This may be done by two different interventions. In the pre-interventional causal graph, we may observe $a \not\perp b$. If we perform $\text{do}(a)$, this dependence still holds. Since we know all influences on a to be cut, only $a \rightarrow b$ is possible. If we choose to perform the hard intervention on b , we may observe an independence $a \perp b$ is created, since the original causal relation $a \rightarrow b$ was eliminated by $\text{do}(b)$. From this change, we can deduce $a \rightarrow b$.

Briefly summarized, we required only one intervention regardless of the variable on which we intervened to identify the present causal relation in a bivariate setup. To identify a causal graph with N variables, Eberhardt et al. [Ebe07] identified the number of required hard interventions to be $N - 1$.

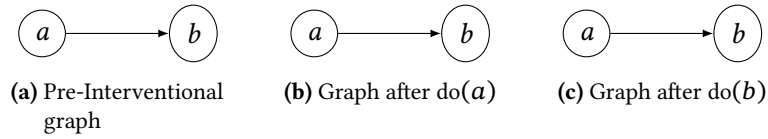


Figure 2.11: Shown is the effect of a soft interventions on a given causal structure. A soft intervention does not interfere with the existing causal structure.

Soft interventions [Ebe14, Cam07, Kor04], also called parametric interventions, introduce a low-intrusive influence on the intervened variable by slightly changing its probability distribution P [Ebe07, Küh11]. As shown in Figure 2.11, the original causal relations are not disrupted. This allows the application in environments where hard interventions are not ethical or possible.

Again, we demonstrate its use on a two variable setup using the variables a and b with $a \rightarrow b$. If the soft intervention $\text{do}(a)$ is performed, then we notice for b that $P(b \mid \text{do}(a)) \neq P(b)$ holds which indicates a causal relation $a \rightarrow b$ must be present.

If we perform $\text{do}(b)$, we notice $P(a) = P(a \mid \text{do}(b))$ holds which indicates $b \nrightarrow a$. Additionally, we observe $P(\text{do}(b)) \neq P(\text{do}(b) \mid a)$ holds indicating that b must be dependent on a . Independently of the variable of intervention, we required only a single soft intervention in a bivariate setup to discover the causal relations. Eberhardt et al. [Ebe07] discovered the number of soft interventions required to identify the causal relations in a multivariate setup to be $N - 1$ for a graph with N variables.

2.4.5 Constraint-based Learning using the PC Algorithm

The Peter Clark (PC) algorithm [Spi00] as shown in Algorithm 1 is the ancestor of constraint-based CSL algorithms. It uses systematic conditional independence tests to infer causal structure information. Testing for conditional independence is known to be inefficient for larger conditioning sets, since it requires the calculation of every possible value assignment for the conditioning set variables [Pet17]. For the PC algorithm, this results in an exponential computational complexity given the number of variables. The algorithm operates in two phases. In the first phase, a graph with fully connected edges \mathcal{E} is created from the inspected variables \mathcal{V} . Briefly summarized, on the inspection of a variable a , variable pairs with its adjacent variables are investigated \mathcal{C} . For each such pair, continually increasing set sizes d of adjacent variables \mathcal{P} excluding b are computed, and for each combination the conditional independence $a \perp\!\!\!\perp b \mid \mathcal{P}$ is tested. If one is present, the corresponding undirected edge between a and b is removed.

Algorithm 1: Identify the causal graph skeleton from data

– PC-Algorithm [Spi00]

Input : Variables \mathcal{V} , Indep. test for $a \perp\!\!\!\perp b \mid \mathcal{P}$

Output: Skeleton of bidirectional edges

```

1  $\mathcal{E} \leftarrow (\mathcal{V} \times \mathcal{V}) \setminus \{(a, a) : a \in \mathcal{V}\}$ 
2 function adj( $a$ )
3   return  $\{b : (a, b) \in \mathcal{E}\}$ 
4  $d \leftarrow 0$ 
5 while  $\exists a \in \mathcal{V}, |\text{adj}(a)| > d$  do
6    $\mathcal{C} \leftarrow \{(a, b) : a < b, b \in \text{adj}(a), |\text{adj}(a)| > d\}$ 
7   for  $(a, b) \in \text{shuffle}(\mathcal{C})$  do
8      $\Gamma \leftarrow \{\mathcal{P} \subseteq \text{adj}(a) \setminus \{b\} : |\mathcal{P}| = d\}$ 
9     for  $\mathcal{P} \in \text{shuffle}(\Gamma)$  do
10      if  $a \perp\!\!\!\perp b \mid \mathcal{P}$  then
11         $\mathcal{E} \leftarrow \mathcal{E} \setminus \{(a, b), (b, a)\}$ 
12      break
13    $d \leftarrow d + 1$ 
14 return  $\mathcal{E}$ 

```

The second phase of the algorithm orients the skeleton's undirected edges to generate a CPDAG by using a given set of orientation rules. The rules are applied on all arbitrary variables $a, b, c, d \in \mathcal{V}$. For some, knowledge of \mathcal{P} from the prior steps independence tests $a \perp\!\!\!\perp b \mid \mathcal{P}$ is required.

- I. Orient $a - c - b$ into $a \rightarrow c \leftarrow b$ if a and b are not adjacent, but share the adjacent variable c and $c \notin \mathcal{P}$.
- II. Orient $b - c$ into $b \rightarrow c$ if $a \rightarrow b$ exists such that a and c are not adjacent.
- III. Orient $a - b$ into $a \rightarrow b$ if the chain $a \rightarrow c \rightarrow b$ exists.
- IV. Orient $a - b$ into $a \rightarrow b$ if $a \rightarrow c \rightarrow b$ and $a \rightarrow d \rightarrow b$ exists such that c and d are not adjacent.
- V. Orient $a - b$ into $a \rightarrow b$ if $a \rightarrow c \rightarrow d$ and $c \rightarrow d \rightarrow b$ exists such that c and b are not adjacent.

The final result of the PC algorithm is a specific equivalence class called Markov Equivalence Class (MEC). Two DAGs G_1 and G_2 are called Markov equivalents, if they share the same probabilistic independencies and conditional independencies [Ver22].

A single CPDAG can represent a MEC of DAGs as we will show below [Spi00]. In Figure 2.12, we demonstrate such a relation.

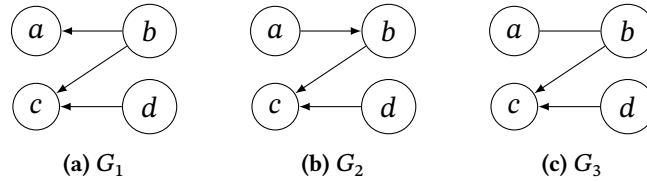


Figure 2.12: The DAGs G_1 and G_2 are members of the equivalence class depicted by the CPDAG G_3 . The edge between a and b in G_3 is known to be present, but can be oriented in either direction.

The DAGs G_1 and G_2 are constrained after the use of a CIT to shown dependencies, because it allowed the identification of the collider $b \rightarrow c \leftarrow d$. As undirected edges of CPDAGs may be directed in the one or in the other direction to form a DAG, G_3 implies G_1 and G_2 given either $a \rightarrow b$ or $b \rightarrow a$ holds.

2.5 Evaluation of Learned Causal Graphs

In general, metrics allow the benchmarking of causal discovery methods on established datasets and thus aid the development of discovery methods. Especially the use of a provided ground true graph has proven to be important to evaluation metrics. By making the found graph comparable to the discovered graph, metrics allow the evaluation of CD methods via their numerical score results [Con19]. As shown in Figure 2.13, the ground true graph may be either provided by an expert. It may also come with benchmarking datasets, which are freely available online.

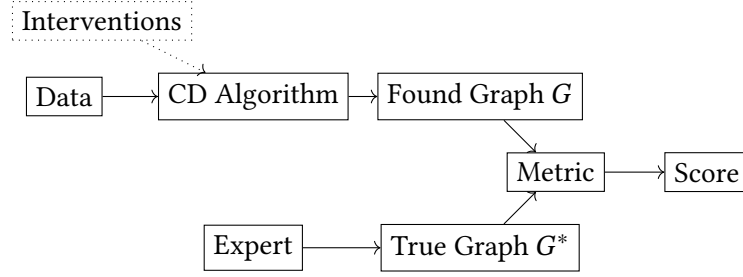


Figure 2.13: The basic approach to evaluate learning method using ground truth

Several metrics that use a ground truth for the evaluation of the discovered graph are currently established in the literature. For their introduction, we use the definition of True Positive (TP), True Negative (TN), False Negative (FN), and False Positive (FP), where positives indicate an edge has been found and negative indicate not edge has been found. The F_1 score [Sas07] considers only the present edges and calculates the harmonic mean of precision and recall when comparing the inferred graph to the ground truth.

$$F_1 = \frac{2 TP}{2 TP + FP + FN} \quad (2.5)$$

The Receiver Operating Characteristic Area Under Curve (ROC AUC) [Faw04, Pet54] requires the True Positive Rate (TPR) and the False Positive Rate (FPR).

$$TPR = \frac{TP}{TP + FN} \quad (2.6)$$

$$FPR = \frac{FP}{FP + TN} \quad (2.7)$$

To evaluate a single learned causal structure, the ROC AUC is measured for an empty graph, for the actual discovered graph, and also for a graph with maximized TPR and FPR. From these measurements, a curve is created plotting the TPR on the FPR and the Area Under Curve (AUC) is calculated. A ROC AUC of one indicates a discovered causal graph that is identical to the given ground truth graph. A ROC AUC of 0.5 indicates the discovery performance to be similar to a random guesser. A ROC AUC of zero indicates the learned graph to be the reverse of the ground truth.

The Precision Recall Curve Area Under Curve (PRC AUC) [Dav06] operates similarly. Precision and recall measurements are taken for an empty graph, the discovered graph and for the graph with maximized TPR and FPR. Then the Precision Recall Curve is constructed by plotting the precision measurements on the recall measurements. Again, the area under the curve is calculated to obtain the PRC AUC value.

The Structural Hamming Distance (SHD) [Tsa06, Aci03] counts the overall number of changes in edges that are required to transform the discovered graph into the ground truth. In the established definition, the allowed changes include only the addition and the deletion of edges.

$$\text{SHD}_{\text{DAG}} = \text{FN} + \text{FP} \quad (2.8)$$

The adapted version for CPDAGs [Tsa06] converts the ground truth and the predicted graph to CPDAGs and then counts the operations required.

$$\text{SHD}_{\text{CPDAG}} = \text{undirected FN} + \text{undirected FP} + \text{FN} + \text{FP} \quad (2.9)$$

To the best of our knowledge, no version is available for MAGs and PAGs.

3 Signal-based Interventions to Gain Structure Constraints

The possibility of learning causal knowledge through intervention and experimentation is one of the advantages of causal learning. In this chapter, we propose a new type of intervention to infer causal knowledge. The main principle of this type of intervention is to add an inconspicuous, but identifiable signal to the intervened variable and to rediscover the signal in dependent variables to gain structure information. The benefits we hope to gain from this particular type of intervention are its low-invasiveness and the learning of structure constraints that contain several variables at once. In the best case, the research will result in a method that can be used during the operation of the system under investigation without requiring a dedicated setup of a test environment. We will give a brief overview of existing intervention approaches in Section 3.1. We explain in Section 3.2 the necessary basics for the design of the signal injection and recovery approaches. In Section 3.3, we present the fundamental concepts to gain causal structure information from the novel type of interventions. Section 3.4 describes in detail two independent methods to inject and recover signals: one of the approaches is based on waveform modulation, while the other is based on frequency modulation. Section 3.5 demonstrates the successful application of the two injection and recovery approaches in artificially generated data. Section 3.6 discussed the usefulness of the developed signal injections and Section 3.7 concludes the chapter.

Parts of this chapter were previously published in [Reh23a].

3.1 Related Work

In Section 2.4.4, we discussed how two different kinds of interventions are established in causal literature [Kor04]. We recall, hard interventions disrupt the original causal flow and the intervened variable is set to a specific value. Soft interventions on the other hand either perform an adaptation on the probability distribution [Ebe07] or, from the viewpoint of functional causal models, adapt the parametrisation of the injection variable's function [Tia01]. Our defined signal injection-based intervention adds an identifiable signal w to the intervened variable v . From the viewpoint of functional causal models, this signal is added to the structure function f_v of the intervened variable v resulting in $f'_v = f_v + w$ without disrupting the original causal relations. Accordingly, it is possible to see signal injections as a special form of soft intervention, since the injection of signals also influences the probability distributions of the injected variables and the variables dependent on them.

The only publication we found to introduce signals to learn causal structure information is the work of Kühnert et al. [Küh19]. They introduced signals to a systems input to analyze if this

contributed to detecting the correct causal relation from input to output. As excitation signals, they chose a white noise signal, a sinusoid signal, a sawtooth wave, an impulse train and a random walk timeseries. They performed them on several dynamic single input single output systems, namely a dead time system, a low pass filter, a nonlinear system and a resonant system. For each is certain that only a causal relation from input to output is present. The bivariate spectral Granger causality [Gra69] was used to identify the causal relation between input and output variables with the help of a surrogate-based independence test. In their experiments, they observed that this style in discovering causal relations highly depended on the choice of exciting signals and on the underlying dynamic system. When tested if the input caused the output, most of the time this causal relation was identified as correct. But if tested for the inverse, several false causal relations were identified in each of the inspected systems and for each injected signal.

Besides this publication, we argue that the basic principle of injecting signals and to gain causal information from their rediscovery is a general established principle in the sciences.

For example in the domain of system identification, it is common to excite a time variant dynamic system using a given signal to learn about its behavior and describe it in the form of equations. The chosen signals are required to excite all relevant frequencies of the system. Common are for example signals as the Dirac impulse or a step response. By observing the corresponding output, we can relate it to the input and learn a transfer function to describe the system's dynamics [Lju05]. The transfer function naturally implies structure as well as parameter information.

In neuroscience, an immersed noise paradigm [Rey03, Lon10] was defined which is related to our work. According to the paradigm, an electric current is added to the membrane of a biological neuron consisting of a noise pattern that mimics synaptic activity and additionally of a systematic signal in form of a current step, an artificial postsynaptic current, a sine-wave signal, or a modulation of the noise amplitude. By measuring the action potentials generated by the neuron in response to repeated injection of signals, conclusions can be drawn about the form of causal relation between the injected signals and the form of action potential [Ili14].

3.2 Background

In the following, we briefly describe all the fundamentals we require to establish the principles of signal-based interventions.

3.2.1 Discrete Fourier Transform

The discrete Fourier Transform (dFT) [Fou88] takes a given signal in its time domain representation and transfers it to the frequency domain. The dFT of a snippet x for a chosen frequency f is defined as follows.

$$\text{dFT}_f(x) = \sum_{n=0}^{N-1} x_n e^{-2\frac{\pi i n}{N} f} \quad (3.1)$$

The Fast Fourier Transform (FFT) [Coo65] is an efficient approach for computing the dFT.

3.2.2 Mueen’s Algorithm for Similarity Search

Mueen’s Algorithm for Similarity Search (MASS) [Yeh16, Mue22] is a pattern matching algorithm that computes a similarity search for a given query sequence q on a given timeseries x , $|x| = n$. First, the query pattern q is inverted and padded with zeros to length n . Then, the z-normalized Euclidean distances are computed in a convolution based manner to grade the distance between q and each possible subsequences of x which may contain q . These temporal ordered grades result in a so-called distance profile for q in x .

For efficiency, they transfer the padded q and x into the frequency domain using FFT, perform multiplication and transfer it back using the inverse FFT. Finally, the results are aggregated. The most similar sequence to q in the distance map is the sequence with the lowest distance value. In general, the algorithm has a complexity of $O(n \log_2(n))$.

3.2.3 Causal Timeseries Dataset Generator

The dataset generator by [Law20] allows benchmarking the discovery capabilities of causal structure learning methods for timeseries data. It can consider desired features for its data generation such as the number of variables and latent variables, the time lag between variables and the nature of added noise.

In general, it first generates a DAG out of the specified set of variables. Then, it extends the DAG to a timeseries graph by discretizing each variable for given temporal sampling steps by considering the specified lags. Then a structural causal model is created by adding structural equations to the original DAG.

To finally create the dataset, first, the noise is randomly sampled from the specified noise distributions for each variable independently. Then, using the generated noise, and the prior created SCM and timeseries causal graph, the variables are sampled in a temporal order. This allows the spread of noise to the dependent variables.

3.3 Causal Constraints from Signal Injections

In this section, we introduce the basic principles of signal injections and how to gain information about causal relations by using them. In its essence, we choose a target variable v_{inj} to inject a signal w to perform our novel intervention. We adapt the do-operator accordingly resulting in $do(v_{inj}, w)$. If the corresponding causal graph is represented as an FCM, the intervention adds the chosen signal w to the function f_v of the injection variable v_{inj} resulting in $f'_v = f_v + w$. As we prefer our intervention to be low-invasive, we prefer w to be of low amplitude to hide it in the natural noise of the investigated system. In the following, we show that w travels to dependent variables of v_{inj} and we will elaborate how we gain causal structure information if we discover w in the observed variables \mathcal{V}_{rec} using our recovery methods $rec(\cdot, w)$. For our approach to

function, we assume the true causal graph to be stationary meaning it is not undergoing foreign changes while we perform our experiments.

Our considerations build on the following assumptions:

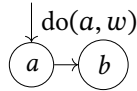
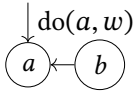
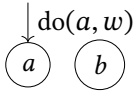
- I. No hidden or outside influence causes w to occur in the observed variables.
- II. We only draw causal deductions from signal discoveries, not from unsuccessful recoveries as we cannot guarantee that a signal is truly absent in a variable. For various reasons, a signal may evade our observation. This may be due to a high damping factor, an immense delay or an unsuitable form and amplitude of the signal.
- III. If we discover an injected signal in a variable, we assume this to be the first occurrence of the signal in the variable.

3.3.1 Spread of Signals in Causal Graphs

In the following, we demonstrate that the spread of an injected signal w depends on the underlying causal graph structure. This is necessary to actually gain information by injecting and recovering w .

Assume a FCM $M = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ of two variables a, b , we perform the intervention $\text{do}(a, w)$ on the variable of injection $v_{\text{inj}} = a$ and meanwhile observe variable b for w ($\mathcal{V}_{\text{rec}} = \{b\}$). In general, several different causal relations are possible between the two variables a and b . For demonstration, let us consider the three cases with corresponding summary graphs shown in Table 3.1. As shown in Case I, after an injection in a , we may observe a remainder of w in b if b is causally depending on a . In Case II and III, b does not contain w .

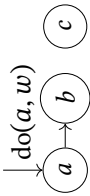
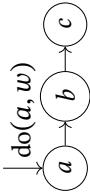
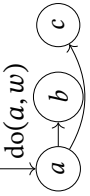
Table 3.1: Given a bivariate FCM, if we inject a signal w in a in a bivariate scenario, our signal w is part of the computation of b only if $a \rightarrow b$. If $a \perp\!\!\!\perp b$ or $a \leftarrow b$ holds, then w is not part of the computation of b .

Case I	Case II	Case III
		
Structure Functions		
$a = f_a(N_a) + w$	$a = f_a(b, N_a) + w$	$a = f_a(N_a) + w$
$b = f_b(f_a(N_a) + w, N_b)$	$b = f_b(N_b)$	$b = f_b(N_b)$
Signal Observation in b		
$f_b(w, N_b)$	-	-

As shown in Table 3.2, this transfers to multivariate setups. We chose for demonstration a graph containing three variables a, b and c . We observe in all three cases that we may find the signals in variables which directly or indirectly depend on the injection variable a , but not in variables which do not depend on it (as for example c in Case I.) Additionally, we observe by comparing

Case II and Case III that we may discover the by structure functions transformed w in the same set of observed variables (b and c) even if the causal graph structure varies. This has effect on the structure learning based on signals constraints because we only consider if a remainder of w is found or not. This will be discussed in the following section in detail.

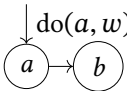
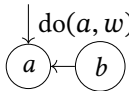
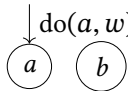
Table 3.2: Given a trivariate FCM, if we inject a signal w in a via $\text{do}(a, w)$, we may observe traces of the signal w to be observed in variables depending on a (Case I) but also in variables indirectly depending on a (Case II and III).

	Case I	Case II	Case III
			
Structure Functions	$a = f_a(N_a) + w$ $b = f_b(f_a(N_a) + w, N_b)$ $c = f_c(N_c)$	$a = f_a(N_a) + w$ $b = f_b(f_a(N_a) + w, N_b)$ $c = f_c(f_b(f_a(N_a) + w, N_b), N_c)$	$a = f_a(N_a) + w$ $b = f_b(f_a(N_a) + w, N_b)$ $c = f_c(f_a(N_a) + w, N_c)$
Observation			
in b	$f_b(w, N_b)$	$f_b(w, N_b)$	$f_b(w, N_b)$
in c	-	$f_c(f_b(w, N_b), N_c)$	$f_c(w, N_c)$

3.3.2 Learning Structure Constraints from Signal Injections

Considering the three possible causal relations and the before introduced assumptions, we can gain constraints about present relations as shown in Table 3.3. If we rediscover the remainder of w in b , then we know in the bivariate case holds $a \rightarrow b$. This may result from the assumption of acyclicity, or be deducted as a is the only potential causing variable, as we assume no foreign influences resulting in the observation of w to be present. Regarding Case II and III, we do not gain knowledge from the absence of w in the observed variables \mathcal{V}_{obs} and thus no constraints can be learned.

Table 3.3: If we inject a signal w in a in a bivariate scenario and we recover the remainder of w in b , then we know b is successor of a . Since we consider the bivariate case, we infer $a \rightarrow b$.

	Case I	Case II	Case III
			
Learned Constraints			
from Recovery	$a \rightarrow^* b$	-	-
from Acyclicity Assumption	$b \nrightarrow a$	-	-
Inferred Constraints	$a \rightarrow b$	-	-
Unknown Constraints	-	$a \multimap b$ $b \multimap a$	$a \multimap b$ $b \multimap a$

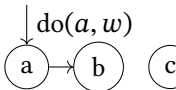
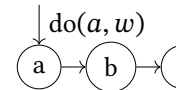
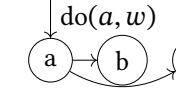
The structure learning is more difficult in a multivariate setup with variables $\mathcal{V} = \{a, b, c\}$ as shown in Table 3.4. In each case, we may identify present paths from the injection variable to the recovery variables. Note that in each case, we can only identify path constraints but cannot identify present edge constraints, since the path of travel of w is not known.

This holds for Case II and III, where w may be observed in the same variables even if the causal structure is different. Since we cannot clearly distinguish the path of travel we may only deduce the paths $a \rightarrow^* b$ and $a \rightarrow^* c$.

Note, we can only learn a path constraint, but no present edge constraint from Case I as well. This is because we do not draw conclusions about the causal structure based on variables where we may wrongly believe w to be absent. As in case I, during structure learning we do not know that c is not directly or indirectly depending on a , but we must consider that the rediscovery of w in c might have failed. As w was discovered in b , we may under the given assumptions discover a path structure $a \rightarrow^* b$, because the causal relations $a \rightarrow c \rightarrow b$ may hold.

Given these constraints, we may infer additional constraints if we assume acyclicity. Corresponding to each edge or path constraint in an acyclic graph, we may deduce the causal relationships in the opposite direction to be absent.

Table 3.4: In a multivariate scenario, we may identify present paths from the recovery of the w and we may identify the corresponding absent edges if we assume the graph to be a cyclic.

	Case I	Case II	Case III
			
Learned Constraints			
from Recovery	$a \rightarrow^* b$ -	$a \rightarrow^* b$ $a \rightarrow^* c$	$a \rightarrow^* b$ $a \rightarrow^* c$
from Acyclicity Assumption	$b \nrightarrow a$ -	$b \nrightarrow a$ $c \nrightarrow a$	$b \nrightarrow a$ $c \nrightarrow a$
Unknown Constraints	$a \multimap b$ $a \multimap c$ $b \multimap c$ $c \multimap b$ $c \multimap a$	$a \multimap b$ $a \multimap c$ $b \multimap c$ $c \multimap b$	$a \multimap b$ $a \multimap c$ $b \multimap c$ $c \multimap b$

3.3.3 Learning Temporal Order Constraints from Signal Injections Causing Unique Signal Recoveries

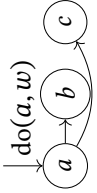
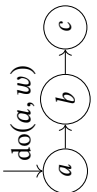
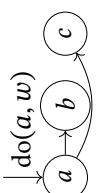
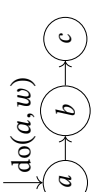
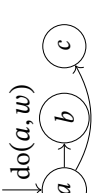
Consider, we retrieve additional temporal information on recovery of a signal. We consider this to be the delay time between the variable of injection and each variable of recovery. These delay times indicate a temporal order between the observed variables. Temporal order was used in the past to draw conclusions about causal constraints. In the following sections, we show how to gain constraints from this temporal order information out of signal injections in addition to the before described constraints of simple signal rediscovery.

For the start, let us assume the timeseries graphs do not contain any cycles and their variables are not autoregressive, meaning the variable do not influence themselves over time. Again, we consider the bivariate case first and use a similar example as before. This time, we inject a signal w in a at the sampling time $t_a = 0$ written as $\text{do}(a, w, t_a)$ and recover the remainder of w with a delay of t_b once in b by performing $\text{rec}(b, w, t_b)$. Since we know $a \rightarrow^* b$ holds, if we discover w in b after injecting it in a . Since no path including other variables may allow $a \rightarrow^* b$ except $a \rightarrow b$, we come to the same conclusion as before. As we know t_a and t_b , we may use the temporal law. It states that the effect of a variable can only occur after its cause $t_a < t_b$ naturally indicating $b \nrightarrow a$. In the bivariate case, this does not prove helpful, but we come back to it in the multivariate case.

If we consider a multivariate case, we may also discover w in variable c with a delay t_c resulting in $\mathcal{V}_{\text{rec}} = \{b, c\}$. Naturally we know $t_a < t_b$, $t_a < t_c$, but what helps us to gain constraints about absent edges is the temporal relation of the variables in \mathcal{V}_{rec} namely t_b and t_c . As Table 3.5 shows, if $t_b < t_c$ holds as in Case I and Case II, we may deduce $c \nrightarrow b$. Same holds for the inverse

in Case III and Case IV. In theory, when it is assured by precise measurement that $t_b = t_c$ holds as in Case V, we may deduce that $b \perp\!\!\!\perp c$ holds as a delay must be present for a causal influence to happen. In Case V, we can thereby deduce that $a \rightarrow b$ and $a \rightarrow c$ must hold. We can apply this temporal law to variable pairs of larger sizes of \mathcal{V}_{rec} to gain absent edge constraints.

Table 3.5: If we assume the summary graphs to be acyclic and assume the variables to not be autoregressive, we may learn the constraints from the example cases as shown. Here, the number below the summary graph variables indicate the order of signal recovery.

	Case I	Case II	Case III	Case IV	Case V
					
Learned Constraints					
from Recovery	$a \rightarrow^* b$ $a \rightarrow^* c$	$a \rightarrow^* b$ $a \rightarrow^* c$	$a \rightarrow^* b$ $a \rightarrow^* c$	$a \rightarrow^* b$ $a \rightarrow^* c$	$a \rightarrow^* b$ $a \rightarrow^* c$
from Acyclicity	$b \not\rightarrow^* a$ $c \not\rightarrow^* a$	$b \not\rightarrow^* a$ $c \not\rightarrow^* a$	$b \not\rightarrow^* a$ $c \not\rightarrow^* a$	$b \not\rightarrow^* a$ $c \not\rightarrow^* a$	$b \not\rightarrow^* a$ $c \not\rightarrow^* a$
from Temporal Law	$c \not\rightarrow b$	$c \not\rightarrow b$	$b \not\rightarrow c$	$b \not\rightarrow c$	$b \not\rightarrow c$ $c \not\rightarrow b$
Inferred Constraints	$a \rightarrow b$	$a \rightarrow b$	$a \rightarrow c$	$a \rightarrow c$	$a \rightarrow b$ $a \rightarrow c$
Unknown Constraints	$a \rightarrow c$ $b \rightarrow c$	$a \rightarrow c$ $b \rightarrow c$	$a \rightarrow b$ $c \rightarrow b$	$a \rightarrow b$ $c \rightarrow b$	- -

3.3.4 Learning Temporal Order Constraints from Signal Injections Causing Recurrent Signal Recoveries

A signal may be discovered several times in a recovery variable. This may be caused by multiple paths leading from the injection variable to the recovery variable, due to cycles in the timeseries graph or it may also be due to autoregressive variables causing themselves over time. As we will show, in such cases it is trickier to draw deductions using the temporal law.

Consider again the bivariate case with variables a , b and $\text{do}(a, w, t_a)$. We may recover w in both a and b multiple times depending on the underlying causal structure as shown in Table 3.6. We can use the temporal law, if all w in the one variable are found either before or after the occurrences of w in the other variable. If w is found interchangeably, we refrain from drawing deductions.

This interchangeable observation occurs in Case II and Case V. We see the interchangeable observations may be due to a cycle ($a \leftrightarrow b$) as in Case II, or due to the causing variable (a) being autoregressive. Since we cannot distinguish such cases, we may not learn constraints using the temporal law on such occasions.

Even if we can use the temporal law, to infer constraints, we still cannot uncover all causal relations. As we can see by comparing Case I and IV, we cannot differentiate if a variable is autoregressive (Case IV) and may have affected itself or if it was caused multiple times by its ancestor (Case I). The exception is if it becomes evident that a variable is autoregressive as in Case III. In it, after the injection in a , w is found again in a before it is found in b .

These principles shown on simple trivariate examples can be transferred to a larger set of variable in which w was recovered, since the featured deduction principles translate to them as well.

Table 3.6: If we allow cycles in the summary graph and allow variables to be autoregressive, we may learn the constraints from the example cases as shown. Here again, the number below the graph variables, indicate the order of signal recovery.

	Case I	Case II	Case III	Case IV	Case V
	 1. 2., 3.	 1., 3. 2.	 1., 2. 3.	 1. 2., 3.	 1., 3. 2.
Learned Constraints					
from Recovery	$a \rightarrow^* b$	$a \rightarrow^* b$ $a \rightarrow^* a$	$a \rightarrow^* b$ $a \rightarrow^* a$	$a \rightarrow^* b$	$a \rightarrow^* b$ $a \rightarrow^* a$
from Temporal Law	$b \not\rightarrow a$		$b \not\rightarrow a$	$b \not\rightarrow a$	
Inferred Constraints	$a \rightarrow b$	$a \rightarrow b$	$a \rightarrow b$ $a \rightarrow a$	$a \rightarrow b$	$a \rightarrow b$
Unknown Constraints	$b \multimap a$	$a \multimap a$ $b \multimap a$	-	$b \multimap b$	$a \multimap a$ $b \multimap a$

3.4 Approaches for Signal Injection and Recovery

In this section, we feature two different approaches which actually perform the signal injection and recovery operation on a system under investigation. The first approach injects and recovers signals with recognizable amplitudes, while the second approach injects and recovers recognizable signal frequencies.

3.4.1 Waveform Modulation-based Injection and Recovery

To perform the signal injection, we generate a specific signal w of preferably unique shape and add it at timestep t_v to the injection variable v by targeted but not disrupting actuation ($\text{do}(v_{\text{inj}}, w, t_v)$).

To recover the signal w in a variable v , we perform the signal recovery step $\text{rec}(v, w, t_v)$. This is done by first linear scaling each timeseries variable $v \in \mathcal{V}_{\text{obs}}$ to a scale of zero to one. Then we independently apply on each observed variable $v \in \mathcal{V}_{\text{obs}}$ the MASS algorithm [Yeh16] and thereby test the signal w against each possible temporal position in v to obtain a distance profile. The lower the distance, the more similar is the amplitude of w to v in the corresponding position and we deem it to be likelier that w is present in v .

3.4.2 Frequency Modulation-based Injection and Recovery

To generate a signal w for this kind of injection, we first sample n values at frequency f_{sample} from a sine curve w of frequency f_{sine} and a chosen amplitude a . This is then multiplied with a Gaussian curve of width $b = 0.5 n f_{\text{sample}}$ and mean $c = \frac{\sqrt{n}}{8}$.

$$w = a \sin(2\pi t f_{\text{sine}}) e^{-\frac{1}{2} \frac{(t-b)^2}{c^2}} \quad (3.2)$$

We assume the sampling frequency to be at least twice as high as the chosen sine frequency $f_{\text{sample}} > 2f_{\text{sine}}$. Finally, we inject the generated signal w by non-disruptive actuation to the chosen injection variable v written as $\text{do}(v_{\text{inj}}, w)$.

For the recovery of w in an observed timeseries variable $v \in \mathcal{V}_{\text{obs}}$, we first sample v at the frequency f_{sample} . Then, v is padded with $\frac{n}{2}$ zero values at each end resulting in v_p , and a sliding Gaussian wavelet (from the injection step) is used to sample snippets from v_p . For each window snippet, we calculate the FFT for the specific frequency f_{sine} we previously used in the injection step. The higher the FFT value at frequency f_{sine} for the snippet, the likelier is the frequency present at the inspected position and thereby the likelier is the presence of the injected signal in v .

3.5 Application Example of Signal Injections

To demonstrate the learning of causal knowledge from signal injections, we adapted the causal timeseries generator framework and applied the frequency and waveform modulation-based injections and recovery approaches. In the following, we describe in detail the choices of parameters and the experiment design followed by the results.

3.5.1 Setup of the Test Environment and Measurements

To benchmark the two injection and recovery approaches, we modified the causal timeseries generator to allow the injection of wavelets in desired variables. In total, ten different causal graphs were generated. On each, the two injection and recovery approaches with the two corresponding wavelets were performed. Each DAG contains five variables, and the numbers of maximum parents and children were set to two. The minimum lag was limited to be larger than the injected wavelets of length $n = 60$. Using the data generator, two independent Gaussian noises with $\mu = 0$ and $\sigma = 0.09$ and $\sigma = 0.07$ respectively were added to each variable independently and spread according to the causal graph.

To evaluation the discovered wavelet positions by the different approaches, we used ground truth given by the data generation framework. As a metric we chose the Precision Recall Curve Area Under Curve (PRC AUC). It provides us the true causal relationships and their temporal occurrences between the variables and we can compute when which variables are affected by the wavelets. The potential wavelet positions are in the case of the MASS distance profile all negative peaks and in the FFT frequency profile all positive peaks. We chose the PRC AUC since it allows the evaluation of how the peak scores help with the distinction between present and absent wavelets without the specification of any specific threshold. Additionally, other than the ROC AUC, it is not prone to imbalances in the label set. From the ground truth, we know such an imbalance between the number of peaks and the actual wavelet positions to be present.

To assess the travel of the signal in each graph, we computed the PRC AUC by considering the ground truth for all variables (PRC AUC (all)) and we computed the PRC AUC given only the ground truth of adjacent variables to the injection variable. As we performed each injection on each of the generated graphs, each of the collected results is the average and standard deviation in PRC AUC of all ten.

For experimentation, we varied the maximum amplitude of each wavelet by choosing $a = 1$ and $a = 0.5$. This allows to inspect how the amplitude size of the signal influences its rediscovery.

In Figure 3.1, we depict some intermediate results. One may observe the Daubechie wavelet and the periodic wavelet with $f_{\text{ sine}} = 100\text{Hz}$ after their injection in the injection variable, their presence in a variable depending on the injection variable and the corresponding profile for the dependent variable indicating the presence or absence of a wavelet. Consider that, the MASS algorithm measures the distance of the signal to the timeseries snippet, this is why lower values are the likelier to be wavelet positions. While the frequency modulation approach identifies snippets with the highest similar frequency as the frequency of the original signal to be wavelet positions.

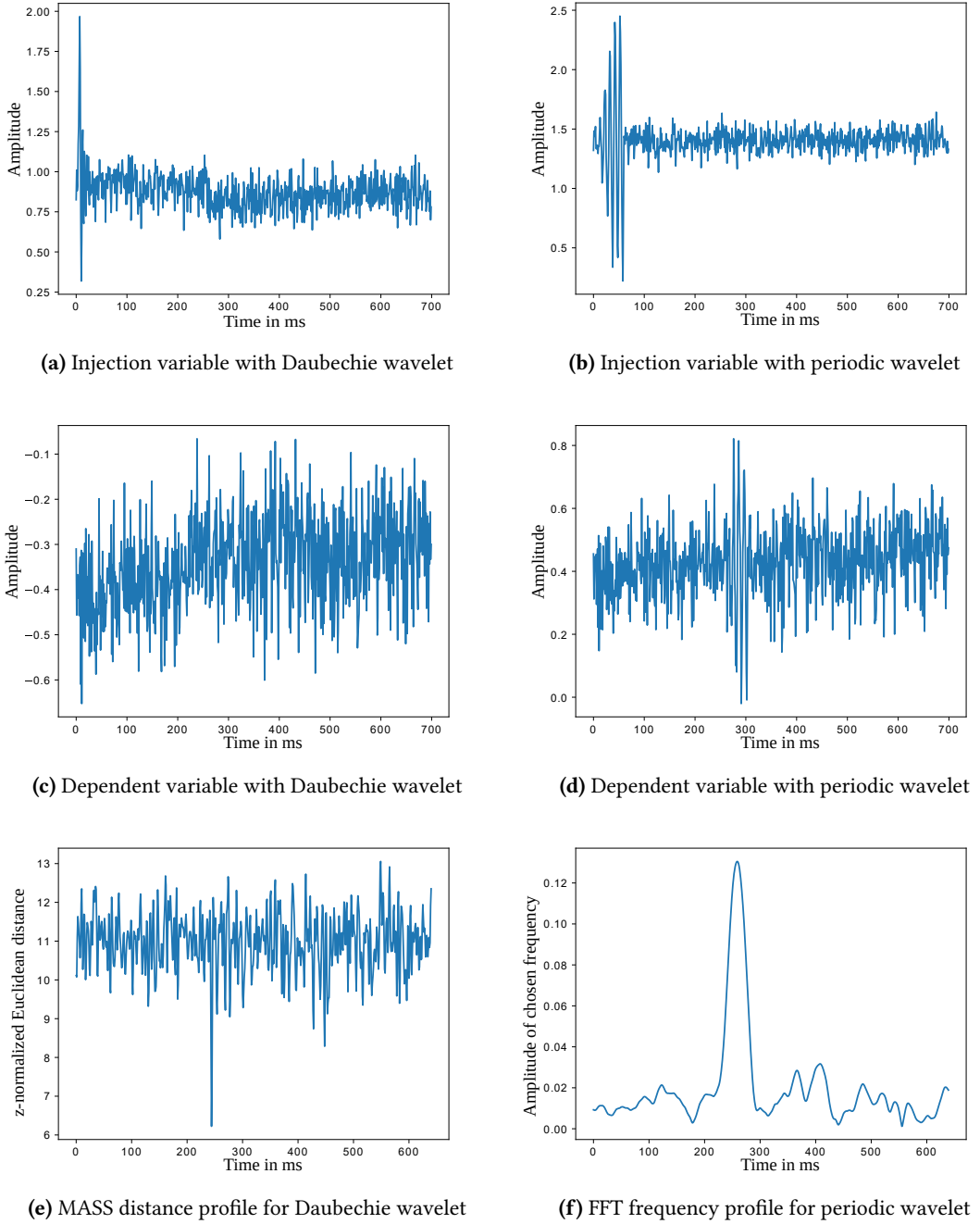


Figure 3.1: An example of how a Daubechie wavelet (a) and a sine-based wavelet with $f_{\text{sine}} = 100\text{Hz}$ (b) and $\alpha = 1$ are injected and how they can travel to an observed variable (c,d) and how they may be rediscovered (e, f)

3.5.2 Setup of the Waveform Modulation Approach

In this approach, we chose a specific signals of unique waveform for injection. We chose the orthogonal Daubechie D30 wavelet [Dau92] as shown in Figure 3.2a. It cannot be described in a closed form, but it can be generated using the cascade algorithm [JIA02] using thirty filter coefficients.

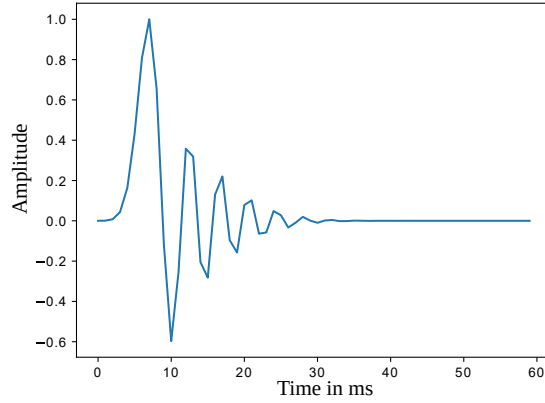


Figure 3.2: Depicted is the resized Daubechie wavelet ($a = 1$) that we chose for amplitude based injection and recovery

We normalized both wavelets by setting their amplitude to one and scaled them up using a chosen scalar a . This is supposed to make the wavelets more comparable in amplitude. By choosing different a over all wavelet sizes, we can experiment with the amplitudes in the rediscovery step.

$$\tilde{w} := \frac{a w}{\max(w)} \quad (3.3)$$

3.5.3 Setup of the Frequency Modulation Approach

We generated two frequency modulated signals with different frequencies $f_{\text{sine}} = 100\text{Hz}$ and $f_{\text{sine}} = 150\text{Hz}$. Both were sampled at a rate (f_{sample}) of 1000 samples per second. To generate both signals, we first constructed a sampling timeseries t with $n = 60$ samples. For the Gaussian function, we had to take the parameters $b = 3$ and $c = 0.96$ for a given number of samples. Figure 3.3 shows the two resulting signals.

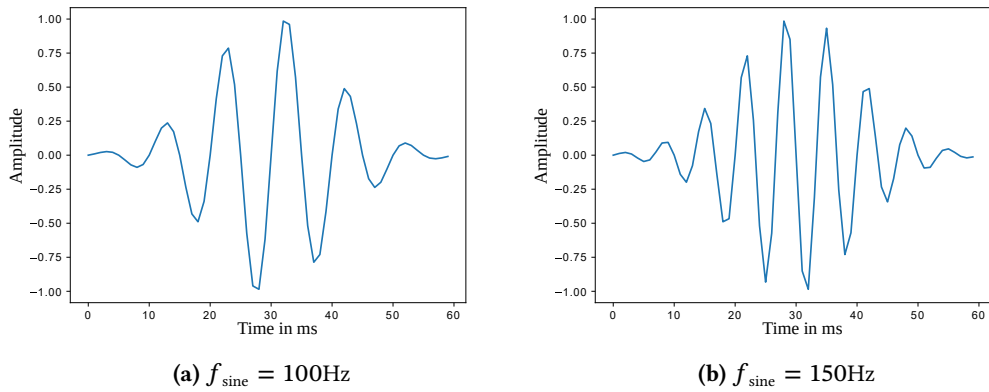


Figure 3.3: Depicted are the two signals we chose for frequency based injection and recovery

3.5.4 Results

The results are depicted in Table 3.7. The higher the PRC AUC scores the better. Regarding the differences between PRC AUC (all) and PRC AUC (adj), we observe independently of the amplitude and the chosen modulation approach the recoveries are better in variables directly dependent on the injection variable.

When comparing the amplitude and the frequency modulation-based approaches, we observe every result of the frequency modulation-based approaches to score higher than the waveform modulation approaches. We assume this to be because of the higher resilience of frequency modulation in regard to noise and signal decay than waveform modulation-based approaches.

Regarding changes in the amplitude, we notice higher amplitudes score higher and better in general. Similarly, in the frequency-based modulation approach, the PRC AUC values for the higher frequency $f_{\text{sine}} = 150\text{Hz}$ score better than all values for $f_{\text{sine}} = 100\text{Hz}$.

In general, we observe all PRC AUC values for signal injections to be high in this artificial testing environment. Even so, no perfect score is achieved, the developed injection and recovery methods provide a foundation to decide if a wavelet is present in a variable or not.

Table 3.7: Results of the signal injection and recovery for different amplitudes. Each entry is the average and standard deviation of ten PRC AUC measurements on ten generated datasets.

Signal Size	Metric	Waveform Modulation	Frequency Modulation	
		Daubechie wavelet	$f_{\text{sine}} = 100\text{Hz}$	$f_{\text{sine}} = 150\text{Hz}$
a = 1	PRC AUC (all)	0.71±0.15	0.87±0.16	0.90±0.12
	PRC AUC (adj)	0.93±0.11	0.90±0.27	0.94±0.41
a = 0.5	PRC AUC (all)	0.66±0.20	0.87±0.16	0.89±0.12
	PRC AUC (adj)	0.87±0.26	0.90±0.27	0.93±0.41

3.6 Discussion

The developed signal injections may prove beneficial as they allow the learning of several constraints at once without disturbing the original causal relations of the investigated system. Our experiments in a generic environment showed that we cannot ideally distinguish between present and absent signals in the observed variables, which may lead to wrong deductions about causal relations.

Regarding the discovery power of signal-based interventions, we can observe a deficiency in comparison to other established intervention types. While hard and soft interventions result in edge constraints in multivariate setups, signal injections mainly result in path constraints even if we apply a large number of signal injections. To make path constraints truly useful, we require always additional edge constraints to gain causal knowledge. Since the primary goal of causal structure learning is to identify edges to gain the true causal graph, we fathom that signal injections are likely to play only a supportive role in future causal discovery. They may

for example prove beneficial in orienting unknown edges left after applying an observational constraint-based structure learning algorithm on given timeseries data.

3.7 Conclusion and Outlook

In this chapter, we proposed a novel kind of intervention that gains causal structure information by injecting an identifiable signal and letting it travel through the process with the original causal information. By observing the other variables for the signal, one may rediscover the injected signal. We elaborated in detail how path constraints and temporal constraints can be generated, if the signal was successfully recovered in an observed variable. For one, we chose only to generate causal information about the presence of a signal, not its absence. Generally, the presence of a signal implies a causal path from the injection variable to the recovery variable. Additionally, we considered the temporal law of causation in our deductions. As we assumed the first discovered signal also to be the first signal present in it, we were able to gain direct edge constraints in such cases.

To perform the actual injection and recovery, we proposed two methods: one uses amplitude-modulated signals, while the other uses frequency-modulated signals. The approaches were exercised on ten different datasets generated by an existing causal timeseries generator framework. We adapted the framework to inject the generated signals and to allow their spread in dependent variables. We applied the methods as specified and experimented with the maximum amplitudes and the form of the signals. We discovered signals of higher amplitudes to be easier discovered by both recovery techniques. In variables directly dependent on the injection variable, it was likelier to distinguish if the corresponding signal is absent or present. In more distant variables, the signals were not discovered as precisely. Still, as this approach is low-invasive and does not interfere with the original causal relations, causal structure learning methods may benefit from it. Consider, that the required probabilistic independence tests of hard and soft interventions also may fail and not perform perfectly, especially if the faithfulness assumption does not hold for the data on the underlying ground true graph.

In future work, one may improve the algorithm and test it on real applications. Further work may for example include the combined use of wavelet transformation instead of the Fast Fourier transformation to discover the wavelet in the frequency-based approach.

4 Efficient Representations of Causal Graph Equivalence Classes

In causal structure learning, several approaches have been developed to gain edge constraints, to learn equivalence classes with them and to represent their equivalence classes as graphs [Mee95, Per17, Aci03]. Commonly, they mainly use edge constraints and do not consider path constraints. Also, they define new labeled equivalence classes and graphs for each representation without considering general approaches to generally encode equivalence classes constrained by edge or path constraints. Such path constraints may be for example learned by using the signal injections described in the previous chapter.

However, a major difficulty in representing equivalence classes is that the number of potential causal graphs in them and thereby the number of constraints grows exponentially with the number of graph variables. When the graphs are represented individually, the storing and handling of such equivalence classes becomes expensive especially when additional constraints are imposed on the equivalence class. To handle larger equivalence classes efficiently, we deem it necessary to develop novel approaches to represent equivalence classes, where at no point the underlying grounded adjacency matrices have to be enumerated.

This chapter introduces several such novel representations, each comes with its advantages. We show that these representations support the addition of more constraints, be it edge or path constraints. Additionally, we demonstrate a novel algorithm PCPC1, which supports the use of path constraints and may use the novel equivalence class representations, while surpassing the original PC algorithm in speed and number of required tests on example datasets.

This chapter is organized in the following manner. We will give a brief overview of relevant related work in Section 4.1. We explain in Section 4.2 relevant background knowledge for this chapter. Then, we present several novel representations of equivalence class representations. In Section 4.3, adjacency matrices are adapted to encode them. In Section 4.4, we show how to use Binary Decision Diagrams (BDDs) for the same purpose. Section 4.5 mainly features how to speed up the creation of such BDDs. In Section 4.6, we feature example applications on an artificial and a real life application case. Section 4.7 mainly features the adaptation of the established Peter-Clark algorithm (PC) and we shown in Section 4.8, how for certain parameter choices the adaptation can exceed the original in time and number of conditional independence tests. Section 4.9 discussed the usefulness of the developed representations and Section 4.10 concludes the chapter.

Major parts of this work are submitted in Pfrommer et al. [Pfr24].

4.1 Related Work

In this section, we look at related work regarding the importance of equivalence classes and their representations in constraint-based learning as well as existing framework which allow the intelligent handling of constraints on equivalence classes.

4.1.1 Constraints in Current Causal Structure Learning

In the established probabilistic causal structure learning, conditional independence tests represent the standard in CSL. Unfortunately, they can only discover causal graphs up to the level of Markov Equivalence Classes [Mee95]. Several additional methods were developed to restrict the MEC with additional constraints. This included for example the use of edge constraints on equivalence classes gained from prior knowledge [Mee95], from interventions in experimental setups [Hau12], or gained from known variable types and their causal relation [Bro22]. As the number of learning methods increases, equally grows the number of graphs types representing the resulting equivalence classes. Perkovic et al. [Per17] summarized the graphs representing a MEC with additional constraints as Maxmally-oriented Partially Directed Acyclic Graphs (RPDAGs). Hauser et al. [Hau12] named the addition of constraints from interventions to CPDAGs interventional essential graphs. Acid et al. [Aci03] developed so-called Partially Directed Acyclic Graphs (RPDAGs) to allow easy inspection of Markov equivalence classes, without having to inspect the entailed DAGs.

Note that each of the shown approaches highly depends on MEC for discovery, while trying to add additional constraints to the equivalence class to find the true causal graph. But the use of probabilistic methods may not always be possible, as they require data which is faithful to the underlying causal graph. The faithfulness assumption may be broken in some systems as described by Andersen et al. [And13]. What we can also observe is that the featured approaches do not provide a general framework to support equivalence classes and the handling of edge and path constraints, which is independent of the specific chosen constraint-learning methods.

Less popular is the use of path constraints in constraint-based causal structure learning. One example is the causal discovery tool CaMML [ODo06] which uses constraints over present edges and present paths from prior knowledge to reduce the graphs in an MEC. Similar to how Borboudakis et al. [Bor12] used prior knowledge of the presence and absence of paths for structure learning. Still, constraint-based learning methods mainly focus on edge constraint, without considering the use of path constraints.

4.1.2 Constraint-Solving Methods

There exist frameworks, which allow the combination of discovered edge constraint sets. The established methods use Boolean Satisfiability Theory (SAT) solvers [Bie09]. They determine an assignment of Boolean values that satisfy given constraints – or proofs that such an assignment cannot exist. This way, SAT-solvers can be used to find valid causal graphs by easily combining multiple constraints in a single satisfiability expression and by trying to solve it.

An example is the algorithm COMBINE [Tri15]. It uses a SAT instance to compute a summary network of constraints given several overlapping observational and experimental datasets. Another is the approach by Hyttinen et al. [Hyt13]. They used SAT solvers to handle constraints from conditional independence tests and prior knowledge.

In general, these SAT-based methods allow the integration of a variety of edge constraints, but they come with the disadvantage that they can only synthesize individual examples of valid causal graphs or show that no such graph can exist. This way, they do not provide an accessible representation of the equivalence class.

4.2 Background

Here we briefly describe essential principles in research, which we will use in this chapter. We will briefly describe the foundations of ternary logic, of binary decision diagrams and the PC algorithm.

4.2.1 Ternary Kleene Logic

Ternary logic introduces a third state besides *true* and *false* called *undefined* or *unknown*. The K_3 Strong Kleene Logic [Kle38] is one approach to capture the respective semantics of such an undefined, additional state. In the following, we depict the defined truth-tables for the ternary Kleene logic:

	<i>false</i>	<i>undefined</i>	<i>true</i>		<i>false</i>	<i>undefined</i>	<i>true</i>
<i>false</i>	<i>false</i>	<i>undefined</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>
<i>undefined</i>	<i>undefined</i>	<i>undefined</i>	<i>true</i>	<i>undefined</i>	<i>false</i>	<i>undefined</i>	<i>undefined</i>
<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>undefined</i>	<i>true</i>
(a) \vee -operation				(b) \wedge -operation			

Figure 4.1: Truth tables of the ternary Kleene logic

4.2.2 Binary Decision Diagrams

BDDs are established in formal verification and allow the representation of Boolean functions $f : \{0, 1\}^M \rightarrow \{0, 1\}$ with M arguments by the use of directed acyclic graphs [Bry92, Ake78]. Each node in the BDD's graph represents one of the input arguments and has two outgoing edges. The edge encoded as high (solid) edge results in the assignment as true, and the edge encoded as low (dashed) edge indicates an assignment with false.

Each path from the root node is required to lead to one of the two terminal nodes \top, \perp . If a path leads to \top , the argument assignments along the path indicate that f evaluates to true. All arguments that are not assigned along the path have no further influence on the outcome. If

a path leads to the \perp terminal node, the arguments assignments along the path indicate that f evaluates to false.

For better understanding, we show in Figure 4.2 an example of a BDD representing the Boolean function $f(x_1, x_2) = x_1 \vee x_2$. The root node is x_1 . It has a solid edge directed at \top , indicating if x_1 is assigned the value true, the result is valid. Its dashed line, indicating the assignment of false, leads to x_2 . Only if also x_2 leads on the high edge to \top and on the low edge to \perp , the BDD represents the condition $x_1 \vee x_2$.

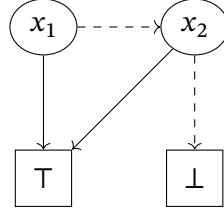


Figure 4.2: A Binary decision diagram representing the Boolean function $f(x_1, x_2) = x_1 \vee x_2$. The root node is defined to be x_1 .

4.3 Adapted Matrices for Equivalence Class Representation

This section features two novel equivalence class representations based on adjacency matrices. Adjacency matrices are a common way to represent the edges of a causal graph. We adapt these adjacency matrices, to increase the number of graphs they can represent.

4.3.1 Ternary Adjacency Matrices

This type of representation requires the definition of a third, undefined state to encode multiple graphs with one adjacency matrix. In the following, we use -1 to indicate if no constraint for a particular edge is given and the underlying causal relation is undefined.

We found that the K_3 Strong Kleene Logic allows the parallel and sequential combination of causal edges and paths using the \vee/\wedge operators. Given a present path $a \rightarrow^* b$ and an undefined path $b \rightarrow^* c$ (where the end variable of one path is the starting variable of the other path), if we assume both to hold $a \rightarrow^* b \wedge b \rightarrow^* c$, according to the \wedge truth table of the Kleene logic, this implies the relationship between a and c is undefined $a \rightarrow^* c$. However, if we have two causal paths with identical start and end variables $a \rightarrow^* b$ and $a \rightarrow^* b$, and we know that both of them hold (written as $a \rightarrow^* b \vee a \rightarrow^* b$, then this implies a present path $a \rightarrow^* b$ according to the \vee -truth table of the Kleene logic.

In the following matrix representations, we use $e_{a,b} = 0$ to indicate the absence of a causal relation from a to b and $e_{a,b} = 1$ to indicate the presence of a causal relation from a to b . Let $\mathcal{A}_3 = \{0, -1, 1\}^{N \times N}$ be the set of all adjacency matrices in the ternary causal logic. Following that notation, the binary adjacency matrices with only defined edges are $\mathcal{A}_2 = \{0, 1\}^{N \times N}$ and $\mathcal{A}_2 \subset \mathcal{A}_3$. Each ternary adjacency matrix $E \in \mathcal{A}_3$ represents an equivalence class \mathcal{A}_2 of the

binary adjacency matrices with which it shares the same 0 and 1 constraints in the same locations. For a number of n undefined edges -1 in \mathbf{E} , the corresponding binary equivalence class contains 2^n binary adjacency matrices each containing a unique combination of 1 and 0 at the prior -1 locations.

To handle equivalence classes and to impose additional constraints, we need to inspect the interaction between adjacency matrices. For demonstration, we use the following five ternary adjacency matrices shown in Figure 4.3. Mind, that the first four matrices are identical in their binary and their ternary representation, but we use them for this example as ternary matrices.

$$\begin{array}{lll} \mathbf{K}_3 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} & \mathbf{L}_3 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} & \mathbf{M}_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ \mathbf{N}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \mathbf{O}_3 = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \end{array}$$

Figure 4.3: As example, we shown five ternary matrices. Each matrix $\mathbf{K}_3, \mathbf{L}_3, \mathbf{M}_3, \mathbf{N}_3$ is a refinement of \mathbf{O}_3 .

The fusing of two ternary matrices is simple if one ternary matrix is a refinement of the other ternary matrix. For demonstration, we use the \cap and \cup notation to hold for ternary matrices, as they represents a set of binary matrices. If we calculate the union of two ternary adjacency matrices \mathbf{A}_3 and \mathbf{B}_3 , the resulting ternary matrix will represent all binary matrices of \mathbf{A}_3 and \mathbf{B}_3 .

A ternary matrix is a refinement, if its binary matrices are a subset of the binary matrices of the other ternary matrix. $\mathbf{K}_3, \mathbf{L}_3, \mathbf{M}_3$ and \mathbf{N}_3 are all refinements of \mathbf{O}_3 . If ternary refinement matrices are joined with their supersizing ternary matrix, the result is the supersizing ternary matrix $\mathbf{K}_3 \cup \mathbf{O}_3 = \mathbf{O}_3$. We can also easily calculate the intersection of two ternary matrices, if one ternary matrix is a refinement of the other $\mathbf{K}_3 \cap \mathbf{O}_3 = \mathbf{K}_3$. If two ternary matrices share a subset of binary matrices, their intersection is the ternary matrix representing this subset. The resulting ternary matrix is a refinement for each of the original ternary matrices.

Unfortunately, we cannot join ternary refinement matrices with diverging sets of binary adjacency matrices if their binary matrices do not include every binary matrix of the supersizing ternary matrix. This is why $\mathbf{K}_3 \cup \mathbf{L}_3 \neq \mathbf{O}_3$ is not allowed to hold, because if true, \mathbf{O}_3 would also entail \mathbf{M}_3 and \mathbf{N}_3 and thus be an over approximation. Instead, $\mathbf{K}_3 \cup \mathbf{L}_3 \cup \mathbf{M}_3 \cup \mathbf{N}_3 = \mathbf{O}_3$ holds, since every binary matrix of \mathbf{O}_3 is included. If we compute the intersection of two distinct ternary matrices, where none is the refinement of the other, we receive always the empty set as a result $\mathbf{K}_3 \cap \mathbf{L}_3 = \emptyset$. As a consequence, we may need multiple ternary adjacency matrices to precisely reflect equivalence classes. In the following, we want to demonstrate that this still leads to an efficiency advantage. We use here a notation $|\mathcal{A}_{3,\mathcal{U}}|$, similar to $|\mathcal{A}_{2,\mathcal{U}}|$, to imply the set of adjacency matrices represent a constrained equivalence class \mathcal{U} . One can observe in Table 4.1, if an equivalence class of graphs with N variables is constrained by simple path constraints like $a \rightarrow^* a$ or $a \rightarrow^* b$, than the number of adjacency matrices in the ternary matrix representations ($|\mathcal{A}_{3,[a \rightarrow^* a]}|$ and $|\mathcal{A}_{3,[a \rightarrow^* b]}|$) are far smaller than the number of adjacency matrices in the binary representation ($|\mathcal{A}_{2,[a \rightarrow^* a]}|$ and $|\mathcal{A}_{2,[a \rightarrow^* b]}|$) for any given $N > 2$ in respect to the specific path constraint. In Table 4.1, we also can observe that the constraint $a \rightarrow^* b$ reduces the numbers of adjacency matrices in an equivalence class more than the constraint $a \rightarrow^* a$ does.

Table 4.1: Number of binary and ternary adjacency matrices required to represent the constrained equivalence classes for growing number of variables N . This table was produced using the Cudd library [Som98].

N	$ \mathcal{A}_2 $	$ \mathcal{A}_{3,[a \rightarrow *a]} $	$ \mathcal{A}_{2,[a \rightarrow *a]} $	$ \mathcal{A}_{3,[a \rightarrow *b]} $	$ \mathcal{A}_{2,[a \rightarrow *b]} $
2	4	1	1	1	1
3	64	36	8	8	40
4	4,096	53	1,152	8	2816
5	1,048,576	538	589,824	53	51,968
6	1,073,741,824	9371	50,921,472	538	25,217,792

4.3.2 Edge Probability Matrices

The previously described ternary adjacency matrices decrease the number of adjacency matrices required to represent an equivalence class. Still, the number of adjacency matrices grows exponentially and for larger equivalence classes it is bothersome to gain an overview of the potential causal graphs, since current methods require to directly assess the individual adjacency matrices. The additional representation we introduce here was developed to provide an overview over the present, absent and unknown causal relations in an equivalence class using a single adjacency matrix $\mathbf{L}_{\mathcal{U}}$. If each DAG in a causal graph is assumed to be equally likely, the matrix entry $l_{a,b}$ indicates formally the probability of an edge $a \rightarrow b$ to be present in an equivalence class \mathcal{U} . Each entry $l_{a,b} \in \mathbf{L}_{\mathcal{U}}$ results from the number of binary graphs where an edge $a \rightarrow b$ is present divided by the total number of binary graphs in \mathcal{U} . If $l_{a,b} = 0$, the edge is absent in \mathcal{U} . If $l_{a,b} = 1$, the edge is known to be present in \mathcal{U} .

The computation of $\mathbf{L}_{\mathcal{U}}$ is given in Algorithm 2 for binary matrices and in Algorithm 3 for ternary matrices. Consider, that the binary algorithm only needs to calculate the set size of given $\mathcal{A}_{2,\mathcal{U}}$ to gain the total number of binary graphs, while the ternary algorithm needs to compute it by summing the number of binary matrices implied by each ternary graph. Regarding the count of $e_{a,b} = 1$ in both matrix sets, the binary algorithm only needs to count each $\mathbf{E}_2 \in \mathcal{A}_{2,\mathcal{U}}$ where $e_{a,b} = 1$, while the ternary algorithm needs to compute for each $\mathbf{E}_3 \in \mathcal{A}_{3,\mathcal{U}}$ the number of underlying binary adjacency matrices 2^k , with k being the count of undirected edges in \mathbf{E} , to then infer the count of binary matrices with $e_{a,b} = 1$.

Regarding the notation, we use the null matrix $\mathbf{0}_N$ given $N = |\mathcal{V}|$. Both algorithms perform in linear time regarding the number of matrices in the given equivalence classes.

Algorithm 2: Computation of the edge probability matrix
for given binary matrices

Input : Equivalence Class \mathcal{U} , Set of binary matrices $\mathcal{A}_{2,\mathcal{U}}$

Output: Edge probability matrix $\mathbf{L}_{\mathcal{U}}$

```

1  $\mathbf{L}_{\mathcal{U}} \leftarrow \mathbf{0}_N$ 
2 for  $\mathbf{E}_2 \in \mathcal{A}_{2,\mathcal{U}}$  do
3   for  $e_{a,b} \in \mathbf{E}_2$  do
4     if  $e_{a,b} = 1$  then
5        $l_{a,b} \leftarrow l_{a,b} + \frac{1}{|\mathcal{A}_{2,\mathcal{U}}|}$ 
6 return  $\mathbf{L}_{\mathcal{U}}$ 

```

Algorithm 3: Computation of the edge probability matrix
for given ternary matrices

Input : Equivalence Class \mathcal{U} , Set of ternary matrices $\mathcal{A}_{3,\mathcal{U}}$

Output: Edge probability matrix $\mathbf{L}_{\mathcal{U}}$

```

1  $\mathbf{M} \leftarrow \mathbf{0}_N$ 
2  $g \leftarrow 0$ 
3 for  $\mathbf{E}_3 \in \mathcal{A}_{3,\mathcal{U}}$  do
4    $k = |\{e_{a,b} \in \mathbf{E}_3 : e_{a,b} = \circ\}|$ 
5   for  $e_{a,b} \in \mathbf{E}_3$  do
6     if  $e_{a,b} = 1$  then
7        $m_{a,b} \leftarrow m_{a,b} + 2^k$ 
8     if  $e_{a,b} = -1$  then
9        $m_{a,b} \leftarrow m_{a,b} + 2^{k-1}$ 
10   $g \leftarrow g + 2^k$ 
11  $\mathbf{L}_{\mathcal{U}} \leftarrow \frac{1}{g} \mathbf{M}$ 
12 return  $\mathbf{L}_{\mathcal{U}}$ 

```

4.4 BDDs for Equivalence Class Representation

Another representation approach to represent equivalence classes incorporates the use of Binary Decision Diagrams. To allow this, the BDD's Boolean function is required to encode the relevant causal edges that need to be present or absent to allow a causal graph to be in the equivalence class for given constraints. We assume \top to indicate such valid graphs, and \perp to indicate graphs outside of the equivalence class. This way, by exploiting structure and redundancy in the Boolean function, the received BDD is typically much more compact than a full truth-table with 2^M entries.

For the efficient construction of such BDDs programming libraries are already available in several languages. To avoid the enumeration of all causal graphs in an equivalence class for the creation

of a BDD, we instead create individual BDDs to represent the equivalence class for each constraint and fuse these BDDs as this operation is supported by the existing libraries. To construct a BDD out of a present or absent edge constrained equivalence class $[a \rightarrow b]$ or $[a \nrightarrow b]$ is trivial, as the corresponding BDD consists of a single node and outgoing edges to the corresponding \top and \perp nodes. The bigger challenge is to compute the BDD for an equivalence class constrained by present or absent path constraints $[a \rightarrow^* b]$ or $[a \nrightarrow^* b]$. In both cases, the equivalence class with all paths of length k from a to b can be constructed recursively from the corresponding equivalence classes with paths of length $k - 1$ according to Equation (4.1).

$$\begin{aligned} [a \rightarrow^k l \rightarrow b] &= [a \rightarrow^k l] \cap [l \rightarrow b] \\ [a \rightarrow^k b] &= \bigcup_{l \in \mathcal{V}} [a \rightarrow^{k-1} l \rightarrow b] \end{aligned} \quad (4.1)$$

Thus, the equivalence class with all causal graphs containing a path between a and b is $[a \rightarrow^* b] = \bigcup_{k=1}^N [a \rightarrow^k b]$.

4.5 BDD Computation Speedup by Rotation

The drawback to encode path constrained equivalence classes with BDDs is that they are far more computation intensive than the encoding of edge constrained equivalence classes. This becomes evident in Table 4.2. We see, that the number of BDD nodes required to encode path constrained equivalence classes is by a magnitude larger than the encoding of edge constrained equivalence classes. If we compare the BDD nodes required to encode an equivalence class constrained by a path $[a \rightarrow^* a]$ to an equivalence class constrained by $[a \rightarrow^* b]$, we can observe that the former requires more than double the amount of the latter. This difference is due to $a \rightarrow^* a$ constraints allowing paths to any variable $v \in \mathcal{V}$ but requiring additional paths $v \rightarrow^* a$ back to the starting variable a .

Table 4.2: Calculation time and number of required BDD nodes for the shown constraints. Temporal measurements were taken by measuring ten times and by calculating the average. The BDD results were produced by cudd library [Som98].

N	$[a \rightarrow b]$		$[a \rightarrow^* b]$		$[a \rightarrow^* a]$	
	BDD nodes	Time in s	BDD nodes	Time in s	BDD nodes	Time in s
3	13	< 0.0001	22	< 0.0001	26	< 0.0001
4	20	< 0.0001	70	< 0.0001	103	< 0.0001
5	29	< 0.0001	292	0.0008	536	0.0007
6	40	< 0.0001	1,657	0.004	2,417	0.006
7	53	< 0.0001	12,422	0.57	26,972	1.29
8	68	< 0.0001	118,835	15.8	304,832	35.85

We achieved a speed up by avoiding the repeated computation of complex path constraints. The according procedure is shown in Algorithm 4. Instead, we use once computed BDD structures of

two chosen edge constrained equivalence classes $[a \rightarrow a]$ and $[a \rightarrow b]$, to piece two more complex BDDs together that represents the two possible path constraints $[a \rightarrow^* a]$ and $[a \rightarrow^* b]$. For this purpose, the variable names are repeatedly exchanged in the edge constrained BDDs using a dedicated swap function and the intersection formed with the path constrained equivalence class under construction. The repeated renaming and reusing of the same edge constrained equivalence class is possible, since for any equivalence class using the same variable set \mathcal{V} and a given ambiguous edge constraint, we can create any other edge constrained equivalence class for the variable set by renaming the BDD nodes accordingly, since the BDD's bare structure is identical. For example, would simply exchange two variables a and b in the BDD node $a \rightarrow b$ by their novel representations a' and b' resulting in $a' \rightarrow b'$. This renaming of variables in BDDs and the forming of intersections between BDDs is supported by many BDD implementations and is computationally cheap.

Algorithm 4: Renaming of BDD node variables of two given BDDs representing equivalence classes

Input : Number of variables $N \geq 3$,

$\text{swap}(f, a, b)$ returns the BDD f with a and b switched

Output: Equivalence classes $[a \rightarrow^* a]$, $[a \rightarrow^* b]$

```

1  $\mathcal{U}_{a,a} \leftarrow [a \rightarrow a]$ 
2  $\mathcal{U}_{a,b} \leftarrow [a \rightarrow b]$ 
3 for  $k \in \mathcal{V} \setminus \{a\}$  do
4    $f \leftarrow \mathcal{U}_{a,b} \cap [b \rightarrow a]$ 
5   for  $m \in \mathcal{V} \setminus \{a\}$  do
6      $\mathcal{U}_{a,a} \leftarrow \mathcal{U}_{a,a} \cup \text{swap}(f, b, m)$ 
7    $g \leftarrow \text{swap}(\mathcal{U}_{a,b}, b, c) \cap [c \rightarrow b]$ 
8   for  $m \in \mathcal{V} \setminus \{a, b\}$  do
9      $\mathcal{U}_{a,b} \leftarrow \mathcal{U}_{a,b} \cup \text{swap}(g, c, m)$ 
10 return  $\mathcal{U}_{a,a}, \mathcal{U}_{a,b}$ 

```

4.6 Application on Example Equivalence Classes

Here, we perform the prior described representations for equivalence classes on a synthetic and a real world example to show their usefulness.

4.6.1 Example Using Generic Data

Assume an equivalence class of acyclic graphs with four variables each is given. We know an edge $a \rightarrow b$ to be present and a path $a \rightarrow^* c$ to be absent. In Figure 4.4, we see the possible representations for this equivalence class. The equivalence class may either be represented by its simple logical representation, by five ternary matrices, by an edge probability matrix or in detail by a BDD containing twenty one nodes and two terminal nodes.

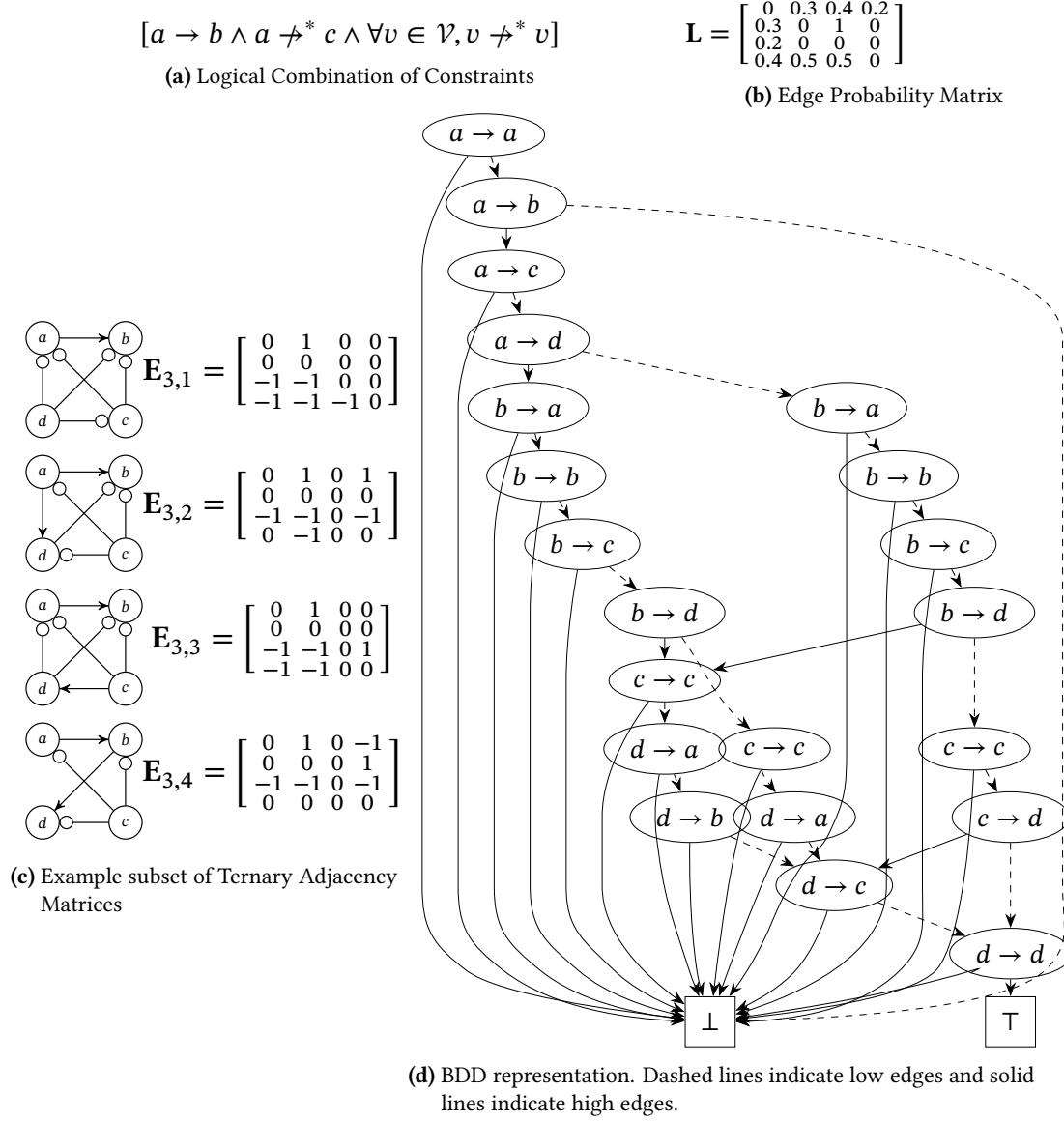


Figure 4.4: Alternative representations of the same equivalence class with four variables.

4.6.2 Example Using Data from Karst Rock

This example demonstrates how the novel equivalence class representations may support the glacier research on the underground waterways in Karst Rock. In the past, researchers injected tracer chemicals into selected springs and aquifers and checked the surrounding waterways for their traces to learn their relations [Sma88]. We use a simplified version of the Castleguard System as shown in Figure 4.5.

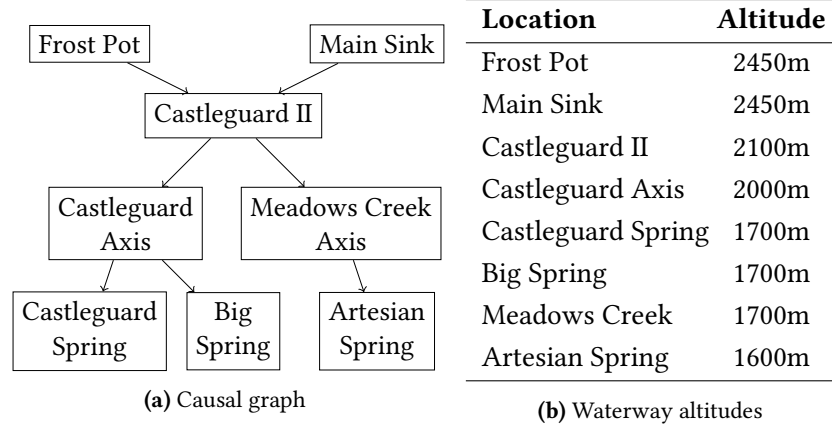


Figure 4.5: Simplified causal graph and altitudes of the Castleguard system.

The injection experiments shown in Table 4.3 were performed one at a time and measurements were taken at the other sites.

We imposed several constraints onto the equivalence class \mathcal{U} . For each injection, we impose a path constraint for each observation site from the injection variable to the observation site. Also, we impose the graphs to be acyclic, and we impose path constraints on the injection sites to not influence any waterways in higher altitudes since we assume underground water only flows downhill. This results in a total of twenty three path constraints. The number of binary adjacency matrices to represent \mathcal{U} is $|\mathcal{A}_{2,\mathcal{U}}| = 121,322,496$. Our novel representation using ternary adjacency matrices allows $|\mathcal{A}_{3,\mathcal{U}}| = 4,432$. While the representation using a single BDD requires 1,113 nodes. The novel overview adjacency matrix $\mathbf{L}_{\mathcal{U}}$ is small enough to be depicted in Figure 4.6.

$$\mathbf{L}_{\mathcal{U}} = \begin{bmatrix} 0 & 0.41 & 0.79 & 0.58 & 0.5 & 0.5 & 0.51 & 0.5 \\ 0.41 & 0 & 0.79 & 0.58 & 0.5 & 0.5 & 0.51 & 0.5 \\ 0 & 0 & 0 & 0.68 & 0.5 & 0.5 & 0.53 & 0.5 \\ 0 & 0 & 0 & 0 & 0.70 & 0.70 & 0.61 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0.35 & 0.25 & 0.55 \\ 0 & 0 & 0 & 0 & 0.35 & 0 & 0.25 & 0.55 \\ 0 & 0 & 0 & 0 & 0.45 & 0.45 & 0 & 0.73 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 4.6: Edge probability matrix of the Castleguard equivalence class

Table 4.3: Listed are the sites of tracer injection and the sampling sites where the tracers were discovered in the simplified Karst Rock example.

Injection Site	Observation Sites
Frost Pot Junction	Castleguard II, Castleguard Axis, Castleguard Spring, Big Spring, Meadows Creek Axis, Artesian Spring
Main Sink	Castleguard II, Castleguard Axis, Castleguard Spring, Big Spring, Meadows Creek Axis, Artesian Spring
Castleguard Axis	Castleguard Spring, Big Spring
Meadows Creek Axis	Artesian Spring

4.7 Structure Learning with the Path-Constrained PC1

Since our representations of equivalence classes perform for both edge and path constraints and allow the efficient representation of the underlying equivalence class during structure learning, we can create learning algorithms which use both constraint types and also the novel representations. We decided to adapt the first step of the PC algorithm (PC1) [Spi00] by changing its skeleton learning step as Algorithm 5 shows. Since the algorithm adapts the first step of the first phase of the PC algorithm (PC1), we call it the Path Constrained Peter-Clark algorithm (PCPC1).

It starts by constraining the equivalence class \mathcal{U} to the set of acyclic graphs via absent path constraints. As long as there exists enough potential parent variable sets \mathcal{P} for a variable a , it generates potential children b of a and tests for the various parent sets if the conditional independence $a \perp\!\!\!\perp b \mid \mathcal{P}$ holds. If a is independent of b for any set of its potential parent sets \mathcal{P} , a and b are conditionally independent and it is imposed on the equivalence class that there may be no path of length K or lower between a and b in \mathcal{U} .

Algorithm 5: Identify the causal graph skeleton from data
– Path-Constraint PC1 (PCPC1)

Input : Variables \mathcal{V} ,

Maximum path length K ,

Indep. test procedure for $a \perp\!\!\!\perp b \mid \mathcal{P}$

Output: Skeleton as set of bidirectional edges

```

1 function adj( $a$ )
2   return  $\{b : \mathcal{U} \cap [a \rightarrow b] \neq \emptyset\}$ 
3  $\mathcal{U} \leftarrow [\forall a \in \mathcal{V} : a \not\rightarrow^* a]$ 
4  $d \leftarrow 0$ 
5 while  $\exists a \in \mathcal{V}, |\text{adj}(a)| > d$  do
6    $\mathcal{N} \leftarrow \{(a, b) : a < b, b \in \text{adj}(a), |\text{adj}(a)| > d\}$ 
7   for  $(a, b) \in \text{shuffle}(\mathcal{N})$  do
8      $\mathcal{B} \leftarrow \{\mathcal{P} \subseteq \text{adj}(a) \setminus \{b\} : |\mathcal{P}| = d\}$ 
9     for  $\mathcal{P} \in \text{shuffle}(\mathcal{B})$  do
10      if  $a \perp\!\!\!\perp b \mid \mathcal{P}$  then
11         $\mathcal{U} \leftarrow \mathcal{U} \cap [a \not\rightarrow^{\leq K} b \wedge b \not\rightarrow^{\leq K} a]$ 
12        break
13    $d \leftarrow d + 1$ 
14 return  $\{(a, b) \in \mathcal{V} \times \mathcal{V} : b \in \text{adj}(a)\}$ 

```

Other than the PC1 algorithm, in case of conditional independence, it eliminates paths instead of edges and thereby it removes more graphs per independence test from \mathcal{U} than the original PC.

When applying PCPC1 and PC1 on existing benchmark datasets, we discovered that the number of required conditional independence tests is significantly reduced as shown in Table 4.4.

Table 4.4: Number of required conditional independence tests for PC1 and PCPC1 on benchmark examples for causal identification. We used the Fisher z-independence test with a threshold of 0.05. The shown values are averages and standard deviation for 20 runs each.

Dataset	N	PC1	PCPC1
Synth5	5	13.0 ± 0.0	10.0 ± 0.0
Coronary	6	87.1 ± 5.6	42.7 ± 1.5
Car	7	114.0 ± 0.0	21.0 ± 0.0
Asia	8	76.8 ± 4.4	59.2 ± 4.5
GMG8	8	109.7 ± 6.4	91.8 ± 5.0

The newly developed PCPC1 contains the original PC1 algorithm as a special case. Since for $K = 1$, the equivalence class \mathcal{U} constructed in our PCPC1 is restricted by removing only direct edges as well. Hence, for the same order in which conditional independence tests are performed, the resulting sets of adjacent variables are identical.

4.8 Speeding up PCPC1

On investigating the temporal performance of PCPC1 in comparison to the PC1 algorithm, we discovered PCPC1 to also exceed PC1 in computation time if the maximum path length K is constrained during the discovery. The upper limit K for the path length lets us accommodate for systems where the strength of causal relations is weakened with increasing path length. Because in such cases, a weak causal effect along a path with many steps could easily remain undetected by a direct independence test between the variables at either end, and the true causal graph would be removed from the equivalence class. Table 4.5 shows the results for various K on two example datasets. The average and standard deviation were taken for ten code executions on each dataset. Consider, the PC1 algorithm is identical to PCPC1 given $K = 1$. We observe for both datasets, that the PCPC1 for $2 < K < |\mathcal{V}|$ requires fewer independence tests and it performs faster than the PC1 algorithm with $K = 1$.

Table 4.5: Execution times of PCPC1 on two example datasets with limited affected path lengths k . Consider, that for $K = 1$ PCPC1 is identical to the PC1 algorithm. Shown are the averages and standard deviations over ten samples each.

k	# Independence Tests	Time in s
1	13 ± 0.0	0.012 ± 0.034
2	13 ± 0.0	0.012 ± 0.002
3	12 ± 0.0	0.011 ± 0.004
4	10 ± 0.0	0.020 ± 0.024
5	10 ± 0.0	0.019 ± 0.004

(a) Synth 5 dataset

k	# Independence Tests	Time in s
1	107.4 ± 0.51	0.085 ± 0.133
2	107.3 ± 0.48	0.074 ± 0.073
3	86.2 ± 1.23	0.071 ± 0.087
4	70.1 ± 0.87	0.065 ± 0.045
5	57.2 ± 1.31	0.106 ± 0.030
6	42.1 ± 1.19	0.200 ± 0.026

(b) Coronary dataset

4.9 Discussion

The shown representations of equivalence classes may be used to encode CPDAGs, PAGs, MAGs and more, if the corresponding edge constraints are applied. Besides edge constraints, we showed that several of our representations may also allow the use of path constraints. In theory, prior knowledge may be integrated easily into learning algorithms using our equivalence class representations, as this only requires the imposing of constraints on the equivalence class before starting the learning process.

In general, our shown representations are not robust against faulty or conflicting constraints, as they impact the representations heavily. This is a common problem of constraint-based learning approaches. In the case of conflicting constraints, each of the representations imply that the equivalence class is empty.

Now, let us consider the equivalence class representations individually. The proposed ternary adjacency matrices may represent equivalence classes with fewer adjacency matrices than by using DAG adjacency matrices. Still, we may require several ternary adjacency matrices to represent an equivalence class and the number of ternary adjacency matrices still results in a huge amount of graphs for a large number of variables.

The BDD representations have the advantage to be human accessible and their logic is already supported by many programming libraries [Som98, Kra23]. Also, BDDs allow easy sampling from the equivalence class, as only a random walk from the BDD’s root node to the \top node is required to receive a causal graph from the encoded equivalence class.

The introduced edge probability matrices may provide a useful overview for present and absent edges in the corresponding equivalence classes. But except this overview, they do not contain any detailed information about them. Consider also, that they deem every present edge in the equivalence class to be equally likely, which does not necessarily match the probabilities implied by the data.

4.10 Conclusion and Outlook

In this chapter, we have introduced several different ways to efficiently represent equivalence classes. This included the use of ternary logic to decrease the number of binary adjacency matrices significantly by using ternary adjacency matrices. Additionally, we developed an edge probability matrix, that provides an overview of the equivalence class as each of its entries indicates the probability of an edge to be present in the equivalence class given that all binary adjacency matrices are equally likely. Additionally, we introduced the use of BDDs to represent equivalence classes.

The use of BDDs is already established in many programming libraries, which allow the automated addition of constraints. With the novel rotation operation for the fast construction of several path constrained BDDs, we can easily decrease the computation cost for their generation. We extended the first step of the common PC algorithm to use path constraints and BDDs. For a given set of test datasets, this resulted in a decreased amount of conditional independence tests and a shorter computation time than the original PC. To run the experiments, we used an Intel Core i7-8565U quad core processor with 8GB of DDR4-RAM and a 512GB SSD.

Future work might entail transferring the path constraint approach and the BDDs to current algorithms that developed out of the PC algorithm, for example, the Peter-Clark Momentary Conditional Independence algorithm (PCMCI) [Run19] algorithm.

5 Scoring Methods for the Evaluation of Discovered Causal Graphs

For the evaluation of CD algorithms, it has become a standard approach to run novel algorithms on popular benchmarking datasets, for which the ground truth is known [Eme23, Men22]. By choosing a scoring method with desired properties, the learned graph can be compared against the ground truth and the discovery capabilities of the algorithm can be assessed by the resulting score [Che22]. We could for example evaluate the findings of the PCPC1 algorithm described in the previous chapter.

In this style of evaluation, the scoring methods and their properties play a vital role. This is why in this chapter we will investigate current scoring methods of CD by assessing their applicability to different graph types learned by algorithms. We highlight desired criteria and inspect how different state-of-the-art scoring methods facilitate these. Finally, we propose a new scoring method called the normed Causal Edit Distance (nCED).

In Section 5.1, we cover related work in the field of CD scoring methods and prior scoring method evaluations. In Section 5.2, we first introduce a new graph notation to unify the several established causal graph types before proposing our new scoring method. In Section 5.3, we first propose several criteria for CD scoring methods to hold, then we inspect those criteria for the different scoring methods. We discuss the new scoring method and the criteria for scoring methods in Section 5.4 and the chapter will be concluded in Section 5.5.

This work is up for publication in [Reh24b].

5.1 Related Work in Scoring Method Evaluations

Since we contribute in this chapter to the evaluation of scores for causal structure learning and also contribute a causal structure learning score, we consider both aspects in the evaluation of related work.

Regarding the evaluation of structure learning scores, several few publication can be found. The work of De Jongh et al. [Jon09] applied the Greedy Thick Thinning algorithm and the Greedy Equivalence Search algorithm on three datasets and used several scores to inspect the resulting CPDAGs and DAGs. The scores included the number of FN edges, the number of FP edges, the sum of undirected and directed TP edges, the number of TP edges, the number of TP edges with correct orientation, the number of edges with incorrect orientation, a weighted sum of FN, FP

and undirected and directed TP edges, a weighted combination of FN, FP, TP, and finally the Structural Hamming Distance (SHD) as the sum of TP and FN. They identified the SHD as a useful score, but they advised against the use of a single score, as the choice of a single score leads to information loss.

The work of Constantinou et al. [Con19] included a general inspection of causal discovery scores. The inspected scores such as the TP, FP, TN, FN, Precision, Recall, the F1 measure, the SHD and the DAG Dissimilarity score [Con21] and discovered imbalances in the score scores. The use of scores based on scoring functions may be biased due to the scoring function itself. Scores using ground truth are reported to be biased by imbalances in the number of edges or independencies in the ground truth. TP, FP, TN, and FN, as well as Precision and Recall, were reported to be biased as each does not integrate enough TP, FP, TN and FN values sufficiently to be independently meaningful.

Of the described publications, none investigated the use of current methods such as the Precision Recall Curve and the Receiver Operator Curve Area under Curve [Pfo19, Pér18]. Additionally, neither of them investigated scores to be applied on learned ancestral graphs.

Regarding related work in respect to specific scores used for structure learning, we can refer to the fundamentals introduced in Section 2.5. Besides them, our novel score has a background in basic graph theory, where it is established to compare graphs according to their structure elements [Gao10, Wil20]. The most popular representative is the Graph Edit Distance (GED) metric [Abu15]. It compares the fit of two given graphs G_1 and G_2 by computing the summed costs overall required edge and node insertions, substitutions and deletions to transform G_1 into G_2 .

As the GED itself is not in use in the current CD, it was not part of our investigation. Instead, we investigate the SHD. It is a descendant of the GED, but other than the GED, it only computes the summed costs for the substitution, insertion and deletion of edges, since the nodes are expected to be fixed. Commonly, the costs for each operation are set to one.

5.2 Definition of the Normed Causal Edit Distance

In the following, we introduce an universal representation to render established causal graphs comparable, to then score them using our novel scoring method. This is followed by an elaboration on choosing the scores parameter, and a short demonstration of its usefulness to employ MAGs.

5.2.1 Universal Causal Graph Representation

To make all the various learned graphs comparable, we transfer them to a unified representation. Consider causal graphs $G = (\mathcal{V}, \mathcal{E})$ with variables \mathcal{V} and edges \mathcal{E} . Here, the edges \mathcal{E} are represented by an adjacency matrix. Each edge $e_{a,b}$ represents knowledge about a present causal relation from variable a to b and also has a counterpart $e_{b,a}$. Edges take on values $e_{a,b} \in \{-1, 0, 1\}$. $e_{a,b} = 0$ indicates the absence of a causal relation from a to b . $e_{a,b} = -1$ indicates the edge is

undirected. $e_{a,b} = 1$ in the adjacency matrix indicates the presence of a causal relation. Hence, for $N = |\mathcal{V}|$ the adjacency matrix is $\mathbf{E} \in \{-1, 0, 1\}^{N \times N}$. The adjacency matrix representation allows cyclic causal relations if not actively prohibited. Causal self-references of variables, the smallest possible cycles, are excluded by $\forall a \in \mathcal{V} : e_{a,a} = 0$.

5.2.2 Calculating the Causal Edit Distance

The Causal Edit Distance (CED) between two graphs G and G^* uses a comparison between the corresponding adjacency matrix elements $e_{a,b} \in \mathbf{E}$ and as $e_{a,b}^* \in \mathbf{E}^*$ follows.

$$\text{CED}(G, G^*) := \sum_{\substack{(a,b) \in (\mathcal{V} \times \mathcal{V}) \\ a \neq b}} f(e_{a,b}, e_{a,b}^*) \quad (5.1)$$

For its computation, we require the following function f using the parameter $0 < k < 0.5$.

$$f(e_{a,b}, e_{a,b}^*) := \begin{cases} 0 & \text{if } e_{a,b} = e_{a,b}^* \\ k & \text{if } e_{a,b} \neq e_{a,b}^* \wedge e_{a,b} = -1 \\ 1 & \text{else} \end{cases} \quad (5.2)$$

The normalized Causal Edit Distance (nCED) additionally requires the division of the CED with the maximum possible editing costs $N(N - 1)$ given $N = |\mathcal{V}|$ variables.

$$\text{nCED}(G, G^*) := \frac{\text{CED}(G, G^*)}{N(N - 1)} \quad (5.3)$$

5.2.3 Analysis of Parameter Choice

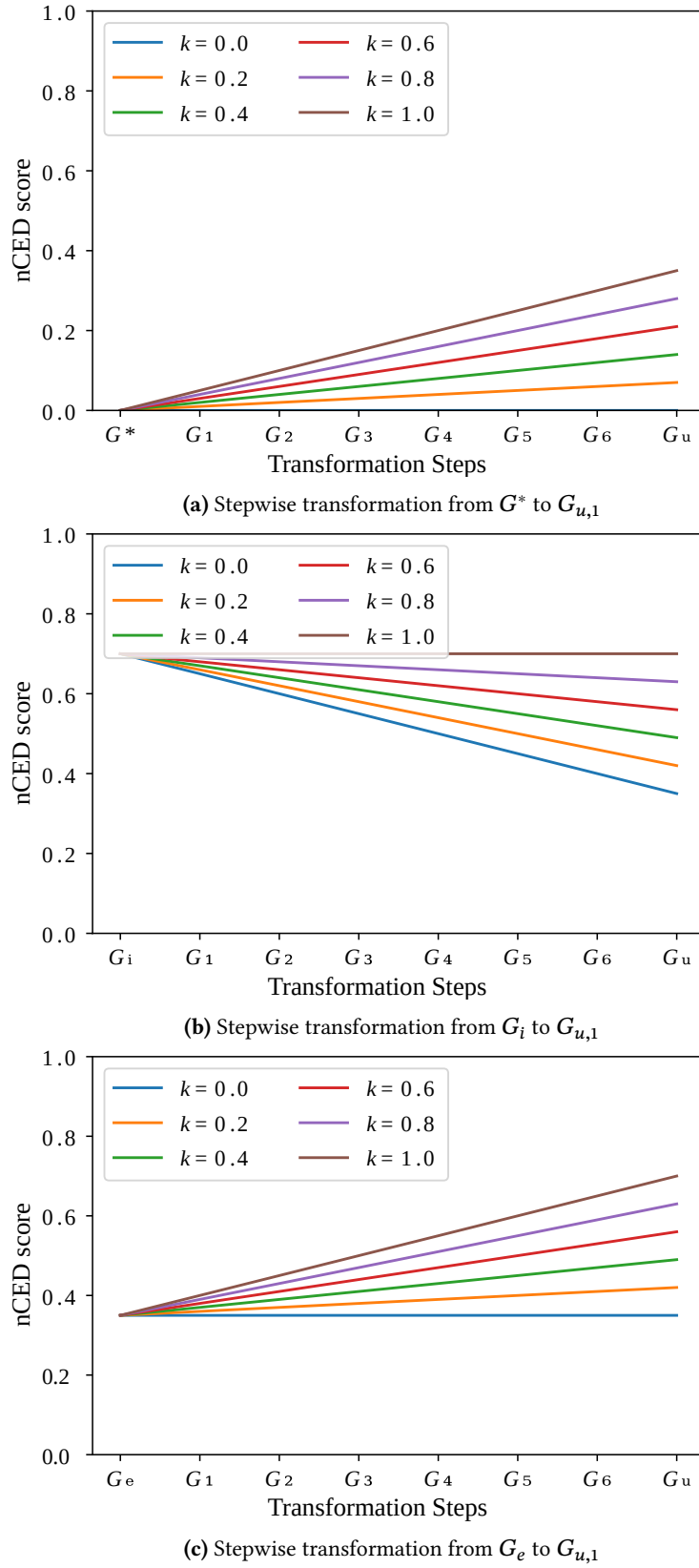


Figure 5.1: Overview of the nCEDs behavior for varying k on stepwise graph transformations of a five variable DAG if the transformation only affects one edge end of an undirected edge.

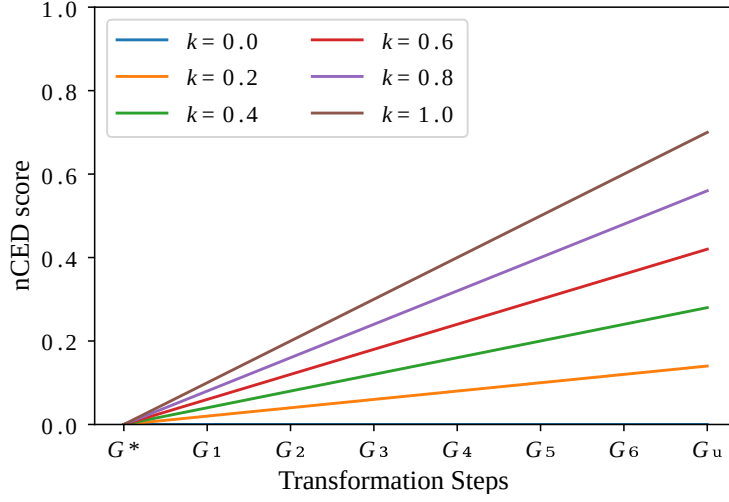
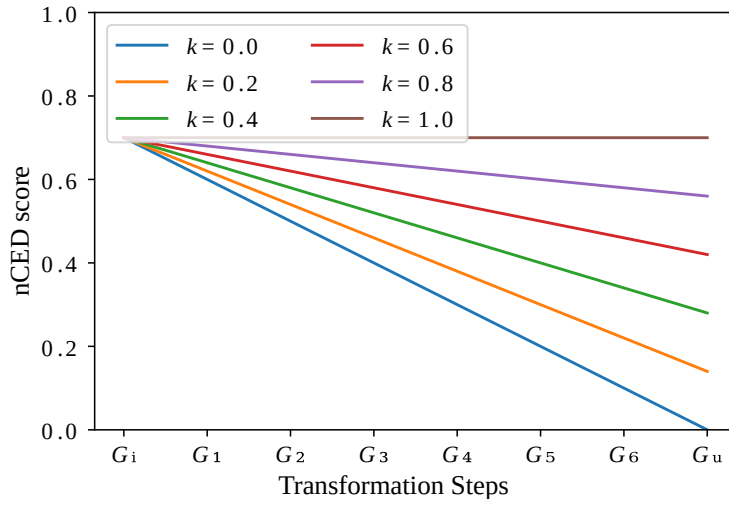
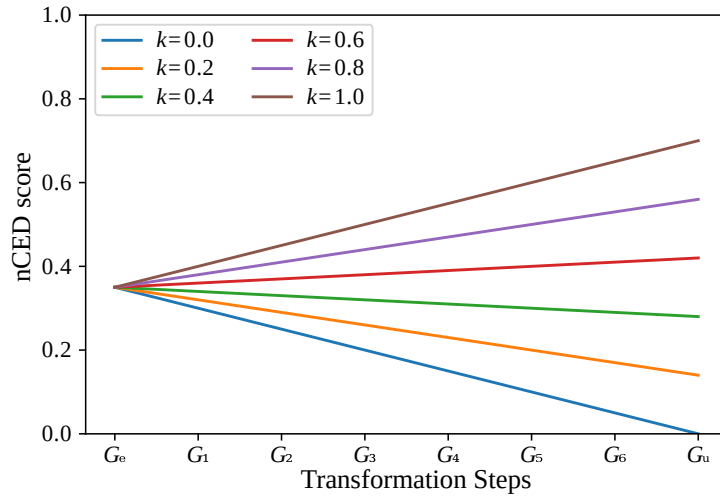
(a) Stepwise transformation from G^* to $G_{u,2}$ (b) Stepwise transformation from G_i to $G_{u,2}$ (c) Stepwise transformation from G_e to $G_{u,2}$

Figure 5.2: Overview of the nCEDs behavior for varying k on stepwise graph transformations of a five variable DAG if the transformation only affects both edge ends of an undirected edge. Since we want the condition $nCED(G_e, G^*) < nCED(G_{u,2}, G^*)$ to hold true, we can deduce from (c) that we need to choose $0 < k < 0.5$

Due to the function f , the algorithm punishes each TP and TN with zero, each undirected FP and FN with k and each FP and FN with one. Given $0 \leq k \leq 1$, k can be freely chosen. Consider, that $k \leq 1$ is required for the normalization step to be valid and $0 \leq k$ for edges to be scored independently.

In the following, we propose our preferences in the choice of k . As undirect TP edges correctly imply the presence of a TP edge, they should be evaluated better than FN edges, but worse than TP edges as they do not imply it with certain orientation.

$$f(e_{a,b} = 1, e_{a,b}^* = 1) < f(e_{a,b} = -1, e_{a,b}^* = 1) < f(e_{a,b} = 0, e_{a,b}^* = 1) \quad (5.4)$$

Likewise, undirected FP edges should be scored worse than TN edges, as they imply faulty information, but they should be evaluated better than FP edges, since the certain discovery of an FP edge is more severe.

$$f(e_{a,b} = 0, e_{a,b}^* = 0) < f(e_{a,b} = -1, e_{a,b}^* = 0) < f(e_{a,b} = 1, e_{a,b}^* = 0) \quad (5.5)$$

Accordingly, the score of an undirected graph $\text{nCED}(G_u, G^*)$ should be better than the score of an empty graph $\text{nCED}(G_e, G^*)$ and better than the score of a graph with flipped edges $\text{nCED}(G_i, G^*)$, but worse than $\text{nCED}(G^*, G^*)$. This should hold for undirected graphs with individual undirected edges $G_{u,1}$ (as is possible in PAGs), but also where opposing edges are undirected $G_{u,2}$ (as is the case in CPDAGs).

We observed the behavior of the scoring method for different k by stepwise transforming G_e , G_i and G^* into $G_{u,2}$, which results in the plots shown in Figure 5.1 and Figure 5.2. One observes $\text{nCED}(G^*, G^*) < \text{nCED}(G_{u,2}, G^*)$ and $\text{nCED}(G_{u,2}, G^*) < \text{nCED}(G_i, G^*)$ to hold for any k . But while $\text{nCED}(G_e, G^*) < \text{nCED}(G_{u,1}, G^*)$ is valid for any k , we see in Figure 5.2c that $\text{nCED}(G_e, G^*) < \text{nCED}(G_{u,2}, G^*)$ only holds if $k < 0.5$. This is why we propose to choose $0 < k < 0.5$.

Alternate desired choices of k might include $k > 0.5$, so that the algorithm treats undirected edges worse than FN and FP. It may be of use if the consideration of undirected edges is to be punished and only the learning of TP and TN is desired.

For $k = 0$, the nCED does not differ between TP and undirected TP edges. This might be of use if the orientations of an edge matters less than the knowledge of its presence.

5.2.4 nCED with Maximum Ancestral Graphs as Ground Truth

As our universal representations allows the nCED to evaluate DAGs, CPDAGs, MAGs and PAGs, and DAGs are a subclass of MAGs, the nCED is also able to use MAGs besides DAGs as G^* . This is because the edge representations of DAGs and MAGs are identical in the universal representation and does not require an adaptation of nCED's definition. In the following, we demonstrate how the novel nCED score can evaluate learned causal graphs using a MAG as ground truth.

For structure learning algorithms, we employed the PC algorithm [Spi00], which learns CPDAGs as results, the Fast Causal Inference (FCI) algorithm [Zha08b] learns PAGs, while the Hill Climb

Search algorithm (HCS) algorithm [Kol09] learns DAGs. For the independence tests, the Fisher z-test was used, and for score-based CD the Bayesian information criterion. Unfortunately, we know of no algorithm which can learn MAGs.

As a benchmark, we used the Asia dataset [Lau88]. It describes the causal relations of visits to Asia and smoking to the corresponding lung diseases and the medical options for identifying the disease. We created a ground truth MAG, by eliminating the smoking variable from the dataset as shown in Figure 5.3. The novel ground truth MAG does not contain it. In its place, it has a bidirected edge implying a latent variable between the bronchitis and the lung infection variable. After learning ten graphs using the introduced learning algorithms, including the example graphs shown in Figure 5.4, we calculated the averaged nCED scores for different k using the created MAG. The results are depicted in Table 5.1.

Table 5.1: Results of the nCED with a MAG as ground truth for graphs learned on the adapted Asia dataset. The lower the score, the better performed the structure learning algorithm.

	nCED (ours)					
	$k = 0$	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
HCS	0.17	0.17	0.17	0.17	0.17	0.17
PC	0.12	0.13	0.14	0.15	0.17	0.18
FCI	0.12	0.14	0.14	0.15	0.17	0.18

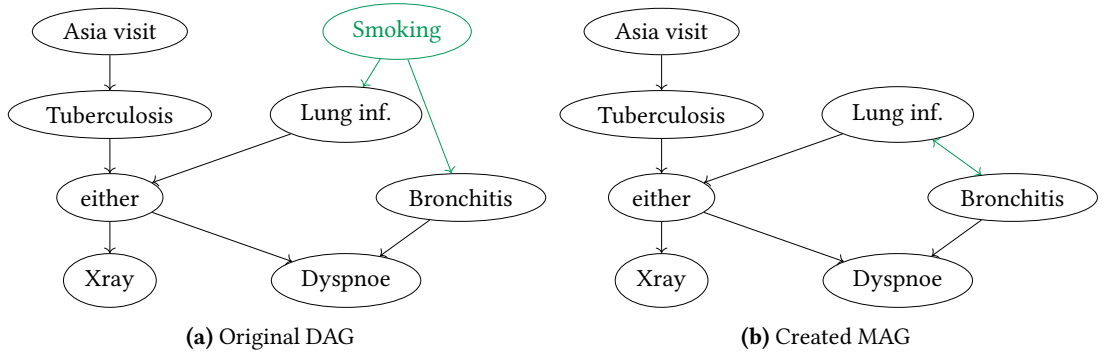


Figure 5.3: The ground truth DAG and the created MAG for the Asia dataset. By eliminating the variable of smoking, we create a hidden confounder between lung infection and the bronchitis variable. The resulting graph is a ground truth MAG. The relevant structure differences are highlighted in green.

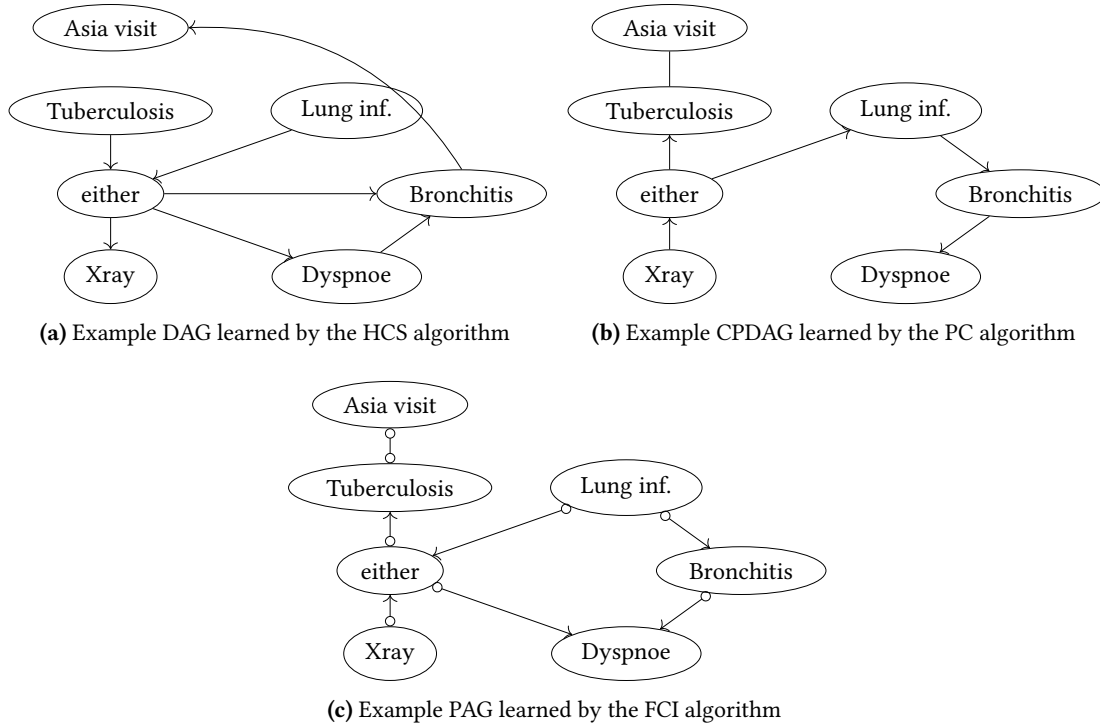


Figure 5.4: Example graphs as they were created by the different learning algorithms when applied on the adapted Asia dataset.

5.3 Benchmark of Learning Scores

We pose a specific set of criteria on any scoring method s that uses solely the ground true graph G^* to evaluate the goodness of fit to any discovered causal graph G .

Lower and upper bound criterion

We prefer the resulting scores of s to be bound by maximum and minimum values. While the use of one bound is common to indicate a perfect match between G and G^* , the use of a second bound is helpful for orientation.

G^* identity criterion

The criterion of G^* identity assumes if $s(G^*, G^*) = s(G^*, G)$ holds, then $G = G^*$ holds.

TP sensitivity criterion and FP / FN sensitivity criterion

Since the true causal graph is assumed to contain only present and absent edges, a scoring method is required to be sensitive to changes regarding TP and FN edges given the present edges of the ground truth, and also to be sensitive to changes in TN or FP edges regarding the absent edges of the ground truth.

Scoring consistency criterion

We try to ensure consistent scoring for variations in TP / FN edges and for TN / FP edges by performing the required changes on G and checking for consistent linear changes in scores. For the TP and FP edges, this entails the continuous increase or decrease of absent edges in G and for

the TP and FP edges the continuous increase or decrease of directed edges in G . This criterion holds if $TP = 0$. This allows comparability between different 'wrong' learned graphs.

Finally, we prefer scoring methods to handle any of the currently established graph types to make the causal discovery scoring methods comparable. This includes DAGs, PDAGs, MAGs and PAGs. Each of the following subsections addresses one of the specified criteria.

5.3.1 Investigation of Applicability

To illustrate the applicability of the existing and the novel scoring method, we demonstrate their use on two example datasets. One is the established Asia dataset [Lau88] with 10.000 data entries with 8 variables each. The second one is the popular Sachs datasets [Sac05] containing 10.000 data entries with 11 variables each. For both, the ground truth is known. Three different CD algorithms were exercised on given datasets. MAGs were excluded as no respective discovery algorithm exists. Again the HCS, the PC Algorithm and the FCI algorithm were employed. All of the examined scoring methods were calculated for the resulting graphs. The scores were collected for 10 learned graphs each. In Table 5.2, we depict the resulting score average. We can observe that most scoring methods are primarily defined for DAGs. One can observe that the PC and FCI algorithm always score worse than the HCS algorithm. If we only inspect scoring methods such as the ROC AUC, PRC AUC, F_1 score, TPR and FPR we would not know if this is due to the reason that they do not consider undirected edges and only evaluate TP, FP, FN, and TN directed edges and thereby leave out information about causal relationships in CPDAGs and PAGs. Regarding the evaluation of discovered CPDAGs, we can see that only our scoring method and the SHD_{CPDAG} is able to evaluate them. As the definition of SHD_{CPDAG} deviates from SHD_{DAG} significantly, and are per se not comparable, we list it in separate columns. All in all, in the established scoring methods, we have a lack of scoring methods which take also CPDAGs and PAGs into consideration and allow a general benchmark between causal structure learning methods without leaving out significant information about causal structure learning. We also show that the nCED may be used as such a benchmark scoring method. In the results for nCED for $k = 0.2$ and $k = 0.4$, we can observe different results for PC and FCI. This shows that the nCED is in fact taking undirected edges into consideration.

Table 5.2: Two examples of how the examined scoring methods evaluate causal graphs learned by the HC), PC and FCI structure learning algorithm. The higher the scores of the ROC AUC, PRC AUC, F_1 , TPR and FPR, the better the algorithm performed. For the SHD and nCED variants, lower scores are better. (* The scoring method only considers present and absent directed edges, unaware of other edge types.)

	ROC AUC	PRC AUC	F_1 score	TPR	FPR	SHD (DAG)	SHD (CPDAG)	nCED ($k = 0.2$)	nCED ($k = 0.4$)
HCS	0.62	0.19	0.33	0.38	0.12	12	-	0.20	0.20
PC	0.47 *	0.13 *	0.06 *	0.06 *	0.12 *	-	11	0.11	0.15
FCI	0.49 *	0.14 *	0.12 *	0.12 *	0.13 *	-	-	0.11	0.15

(a) Asia dataset

	ROC AUC	PRC AUC	F_1 score	TPR	FPR	SHD (DAG)	SHD (CPDAG)	nCED ($k = 0.2$)	nCED ($k = 0.4$)
HCS	0.72	0.33	0.51	0.53	0.09	17	-	0.21	0.21
PC	0.47 *	0.13 *	0.06 *	0.06 *	0.12 *	-	10	0.22	0.26
FCI	0.57 *	0.16 *	0.25 *	0.29 *	0.16 *	-	-	0.23	0.26

(b) Sachs dataset

5.3.2 Experiment on Scoring Consistency and Sensitivity

For this experiment, we considered a generic DAG with $N = 5$ and containing the maximum number of allowed directed edges resulting in seven directed edges and thus seven possible random structure changes. To inspect the scoring behavior regarding TP / FN consistency, we stepwise changed the true DAG G^* into an empty DAG G_e randomly eliminating a TP and adding an FN one by one. Regarding consistency in TN / FP direction, we stepwise changed an empty graph G_e into an inverse graph G_i by randomly adding FP-directed edges directly opposing the edges of the ground truth. The score results are shown in Figure 5.5.

We observe in Figure 5.5a that the PRC AUC is sensitive, but does not perform consistently linear regarding constant changes in TP and FN edges. The FPR has a gradient of zero, as it is insensitive to the performed changes. Regarding sensitivity in TN and FP edges in Figure 5.5b, the TPR, F_1 and PRC AUC show non-responsive by having a gradient of zero. The only scoring methods sensitive to changes in both individual trials are shown to be the ROC AUC, the SHD and the nCED.

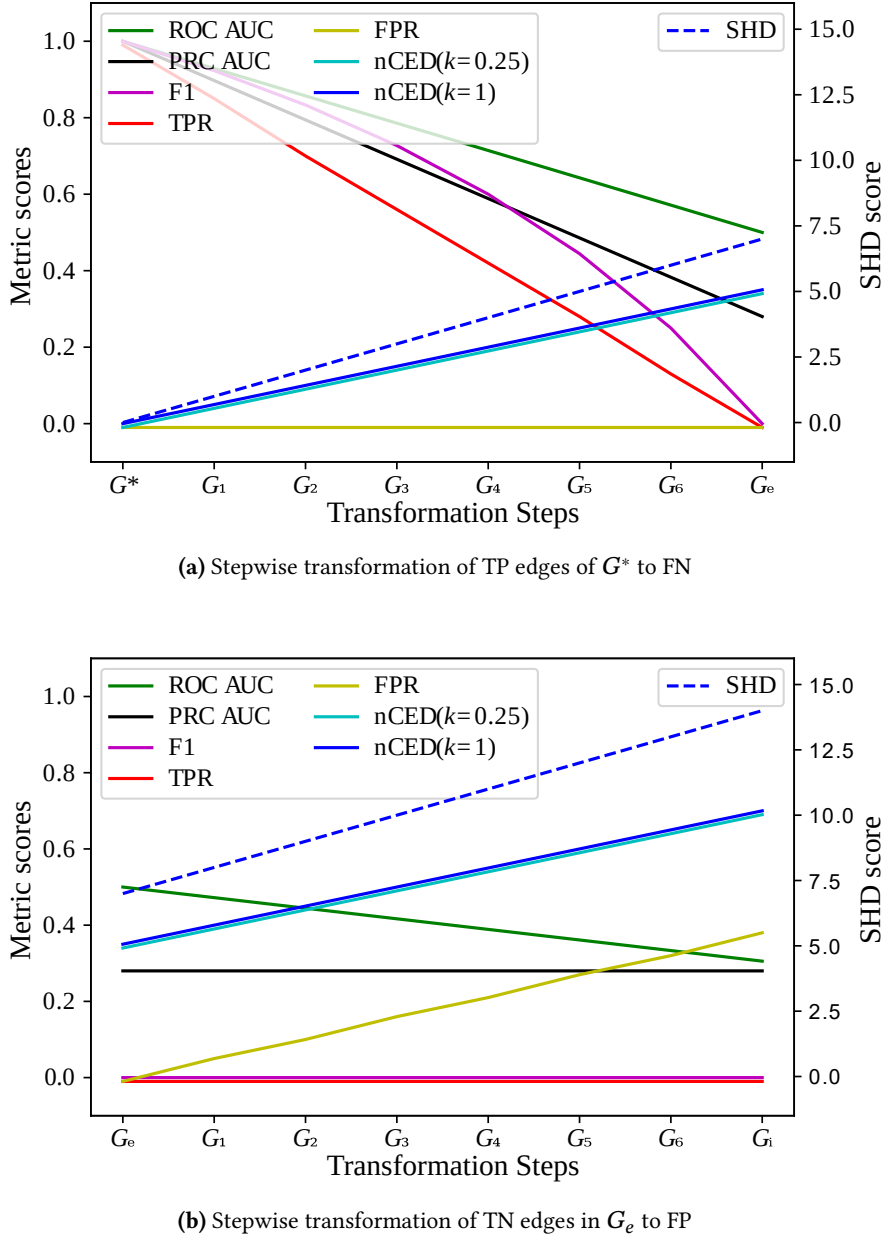


Figure 5.5: Overview of the behavior of the scoring methods given the sequential use of random structure changes on a five variable DAG. The scale of the SHD is shown on the plot's right side. We observe some scoring methods to have a gradient of zero and thereby to not be responsive to the induced changes. Of the responsive scoring methods, we observe most perform consistently indicated by a linear increase or decrease.

5.3.3 Elaboration on Bounds and Normalization

To inspect the bounds of each scoring method, we inspected the definition of the scoring methods and created an overview shown in Table 5.3. Only the SHD comes without an upper bound. All other scoring methods come with an upper and lower bound and are therefore considered to be normalized. While most scoring methods use sums of FN, TP, FP or TN, but never all

of them together, the nCED uses the number of graph variables for normalization. Mind, that $N = TP + TN + FN + FP$ holds and that is why the nCED considers each for normalization.

Table 5.3: Bounds and normalizability of the inspected scoring methods

	Lower bound	Upper bound	Normalizing factor
ROC AUC	✓	✓	$TP + FN$ and $FP + TN$
PRC AUC	✓	✓	$TP + TN$ and $FP + TP$
F_1 score	✓	✓	$\frac{TP}{TP+TN}$ and $\frac{TP}{FP+TP}$
TPR	✓	✓	$TP + FN$
FPR	✓	✓	$FP + TN$
SHD (DAG)	✓	✗	-
SHD (CPDAG)	✓	✗	-
nCED (ours)	✓	✓	$N(N - 1)$

5.3.4 Elaboration on G^* identity

The criterion states that besides G^* no other graph G should exist for which the inspected scoring method evaluates $s(G, G^*) = s(G^*, G^*)$. For this investigation, we tried to find contradicting evidence for each scoring method regarding each criterion.

SHD_{CPDAG} transforms the ground truth DAG to its CPDAG representation. This transformation is commonly surjective as unambiguous edge orientations are removed from the graph and thus several DAGs can share the same CPDAG representation. This is why the G^* identity criterion does not hold for the DAGs sharing the CPDAG representation with G^* .

Several scoring methods, such as the F_1 score, PRC AUC, ROC AUC, TPR and FPR, do not consider undirected edges and they count them as absent edges. This is why the criterion does not hold for graphs identical to G^* , but with undirected edges instead of TN edges.

For $k = 0$, the nCED evaluates undirected edges as identical to TP edges. Thus the criterion does not hold for graphs with arbitrary undirected TP and TP edges.

5.4 Discussion

First, let us reflect on the new nCED scoring method. As we have shown, it allows the benchmarking of different causal graph types. For this purpose, it makes use of an universal representation that simplifies several aspects of causal graph to make them comparable. For example, it considers undirected edges in CPDAGs and PAGs as identical, which is not actually the case. CPDAGs assume undirected edges to be oriented either in the one or the other direction, while allow bidirected edges PAGs. Similarly, a directed edge of CPDAG or DAG is treated as identical to a PAG's directed edge part of a bidirected edge indicating a latent variable.

The scoring method also comes with a parameter k , which may allow causal discovery methods learning undirected edges not to be at a disadvantage from the start in comparison to algorithms

learning DAGs. We proposed a range for k to meet our requirements, but still a suitable value for k needs to be found by empirical investigation.

We have also shown that the nCED may not only use DAGs, but also MAGs as ground truth. To our knowledge, nCED is the only scoring method that is capable of this. Potentially, this may contribute to the development and benchmarking of more discovery algorithms learning MAGs. We argue that DAG- and CPDAG-learning algorithms may still be evaluated if a MAG is used for ground truth, but essentially they may not achieve optimal scores, since they cannot identify bidirected edges or latent confounders, hence they are likely to be punished. (For the special case of $k = 0$ and for specific placements of undirected edges in the learned graph, the ideal score may still be achievable.)

In the inspection of established CD scoring methods, we found the G^* identity criterion and the sensitivity criteria to be essential. Regarding the sensitivity criteria, we used a generic DAG with five variables and repeatedly applied structural changes to it by adding and deleting directed or undirected edges. With five variables the causal graph used for inspection is small, but we argue that the overall behavior of the different observable scoring methods is transferable to applications with more variables, given the ground true graph has as well $\frac{N(N-1)}{2}$ edges.

Besides that, we do not deem our set of criteria to be complete. Instead, we advice to consider and propose more relevant criteria which are essential for causal structure learning scoring methods. This may include for example the robustness of a scoring method against imbalances in the number of present and absent edges, as inspected by [Con19].

5.5 Conclusion and Outlook

In this chapter, we investigated metrics for causal structure evaluation using ground true graphs. We set up a new set of criteria we deem useful to evaluate causal structure metrics using ground true graphs. Additionally, we developed a uniform representation of established causal graphs to make them comparable and developed a new metric that allows the comparison between DAGs, CPDAGs and PAGs and can also flexibly adapt to preferences regarding undirected edges. We inspected the developed criteria for the existing and the novel metric and demonstrated their application on example datasets for comparison. We discovered that many existing metrics only exist for DAGs, and few for CPDAGs. We found no metric for PAGs. Of the existing metrics, we found none that makes the different discovered graph types comparable. We found several metrics, that only consider directed edges and their presence and absence. For some metrics, their ideal score did not guarantee the discovered graph to be identical to the provided ground true graph. As these metrics are used to benchmark and optimize causal structure learning methods, we deem it essential to know metrics in detail to know what they allow optimization for.

6 Root Cause Analysis using Causal Graphs

To assure reliability, efficiency and safety of a system, the discovery of faults, of their corresponding root causes and of potential treatments to eliminate them is essential. An anomaly or fault is commonly defined as an uncharacteristic deviation of a standard value in a system [Sch97]. Such an anomaly may lead to malfunction or failures if not treated. Primarily, the domain of anomaly detection focuses on the discovery of them in data. Root causes are the undesired phenomena causing such anomalies. They are mainly in the focus of the domain of Root Cause Analysis (RCA) and diagnosis. If the root cause analysis step is successful, the domain of adaptive fault management may propose treatment actions to avoid dangers and performance degradation in the given system.

In this chapter, we mainly focus on contributions to the domain of root cause analysis. Several steps have been shown to be essential in RCA [Roo04]. First, information is collected concerning the anomaly. Then, the causal relationships that led to these anomalies are identified followed by the identification of the root cause. Some RCA methods use causal knowledge to perform these steps. This is natural, since causal research inspects in depth the interplay between the variables in given data and allows the identification of causes and effects [Pea10]. The spread of anomalous information in any system has to adhere to the laws of causality and also the nature of the relation between root cause and anomaly is causal.

In this chapter, we propose two approaches which score potential root causes according to their similarity to detected anomalies based on a given summary causal graph. The basic approach uses simple variable sets to match potential root causes. Its extension also considers coarse temporal information for this purpose. Figure 6.1 shows the general setup of both approaches. The areas we contribute to are the preprocessing of given annotated timeseries graphs and the combination of the resulting information with results from anomaly detection to score potential root causes. We neither contribute to the domain of root cause treatment generation nor to the domain of anomaly detection. For a fundamental insight in the latter, we refer to existing work as summarized by Chandola et al. [Cha09].

As root causes, we consider known influences which are unobservable but can be identified according to their effects in the observable variables. As we show in the related work section, this aspect is one of the main differences to existing literature.

The chapter is organized in the following manner. In Section 6.1, we will show related work in RCA and in causal-based root cause analysis. Section 6.2 described the invented RCA algorithm for use on discrete data. Section 6.3 features an adaptation of the algorithm to employ it on time

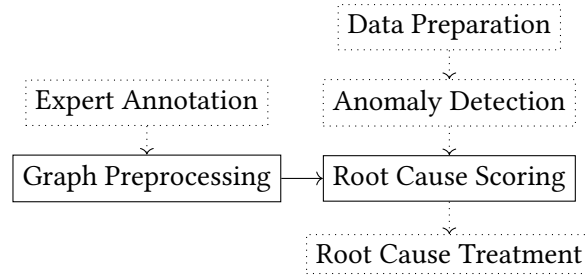


Figure 6.1: A general overview of the steps required for this RCA algorithm. We point out the steps we contribute to in this chapter via solid frames.

series data of production processes. Section 6.4 features the results of the adapted algorithm when it was employed in a robotic gripping process. In Section 6.5, we discuss the developed approaches and conclude the chapter in Section 6.6.

Parts of this work were published in [Reh23b]. Supplementary parts are up for publication in [Reh24c] and [Reh24a].

6.1 Related Work

Existing approaches for root cause analysis and diagnosis have been classified in model-based [Wil76, Ven03a, Ven03c, Ise97] and process history-based approaches. The latter use for example statistical classifier, neural networks and the PCA algorithm to infer root causes from data [Ven03b]. The former, model-based approaches have been divided in qualitative and quantitative models. While quantitative models [Ven03c] include the use of, for example, state or output observer [Gar97] or parity space-based approaches [Pat91], qualitative models [Ven03a] include the use of causal models in the form of, for example, fault trees and directed graphs allowing cycles [Iri79]. In the past, such causal graphs have shown to be gained easily from the process equations or from prior knowledge [Ume80]. [Cha90] introduced several techniques to simplify causal graph structures to improve their use in fault diagnosis.

In more recent literature, several publications advance the use of causal models to infer root causes. Several approaches learn a certain normal behavior of a system from data, to then identify anomalous behavior as the root cause for faults in a system. Yang et al. [Yan22] published a general framework to detect anomalies and to identify root causes in multivariate time series. They learned the causal graph from the data, to then identify modules in the investigated system and performed anomaly detection on each separate module. The root cause is supposed to be the faulty module. Koutroulis et al. [Kou22] combined causal graphs and anomaly detection methods to robustly detect anomalous cyber security attacks in the control system of a water treatment testbed. First, they trained causal models with attack free data offline on the sensor measurements. Then, in an online application, they used the SCMs to compute the prediction error to identify critical influences between variables in the data. As root cause candidates, they identified the sensor variables associated with the attacks. In a similar manner, Shah et al. [Sha18] investigated a semiconductor plant, examining the interactions of system components and their change over time. They learned time series graphs based on recurrent neural networks and

converted them into simplified, directed weighted graphs representing the system. Then, graph similarity techniques were applied to compare the graphs and thereby detect anomalous graphs and thereby changes in the overall plant behavior. The detected change was detected to be the root cause. The framework RootCLAM by Han et al. [Han23] learns a Variational Causal Graph Autoencoder from normal data to then identify in anomalous data the responsible variables in which the presence of a foreign influence (the root cause) is evident. We argue, while these methods have the advantage to learn their causal model from data, they refer to the detected anomalous observations as root causes, without performing deductions about what exactly the hidden influence behind the anomalous observations may be.

We found RCA approaches in literature, which use prior knowledge on root causes in combination with causal graphs to identify their influence. The work of Agrawal et al. [Agr16] used individual Bayesian Networks (BNs) for each type of root cause to diagnose their presence in a coal-fired power plant. Each BN was constructed by experts to contain relevant evidence variables leading to the root cause variable as sink node. Based on fixed thresholds, the variables in the evidence was detected to be either normal or anomalous, accordingly the probability of the root cause variable could be derived. Here, we argue that Agrawal et al. did not represent the true causal relationship of a system, since naturally the evidences are caused by the root cause. Also, they did not inspect the temporal occurrence of evidences to draw their deductions. The procedure EasyRCA by Assaad et al. [Ass23] uses a given acyclic summary causal graph (ASCL) to identify root causes in multivariate timeseries data. According to their work, root causes may be variables with anomalies, whose first appearing anomaly might not have been propagated by any other causing variable under the assumption the anomaly may not precede its cause in time in the ASCL. Additionally, root cause variables are also identified by comparing the adapted total effect estimations on direct relations of a normal and an anomalous regime to detect variables affected by foreign influences. Other than in our work, the summary graph is required to be acyclic. Also, their main focus is on discovering the influences of root causes, while we assume the root causes to be known beforehand.

We also briefly refer to related approaches, which use causal knowledge without constructing causal models. The work of Kühnert et al. uses the definition of causal strength employed by the Granger causality [Gra69] algorithm to measure the causal strength between variable pairs. They identify variables with the most impact on adjacent variables to be the root cause. This was demonstrated on a simulated stirred-tank reactor and on a laboratory plant [Küh13]. Diedrich et al. [Die22] proposed two approaches, an uninformed and an approach informed by residual values. Both learned propositional logic rules using Granger causality from data for the use with consistency-based diagnosis algorithms. This was demonstrated on the Tennessee Eastman process, on a benchmark with systems using multiple-tank examples, and on data from an industrial packaging robot. Of the two approaches, both have the advantage to require minimal prior knowledge. However, as they do not construct a causal graph, they entail few detail about the systems setup, which is helpful to encode system knowledge, may provide intuitive visuals for system monitoring [Hue20] and may provide information about the anomalies propagation path.

6.2 Basic Algorithm Setup

The developed RCA algorithm scores known root causes according to their fit to anomalies detected in advance. It requires a provided causal graph $G = (\mathcal{V}, \mathcal{E})$ which should also contain a set of potential root causes $\mathcal{V}_R \subseteq \mathcal{V}$ and contain information about observable variables $\mathcal{V}_M \subseteq \mathcal{V}$. It may be provided by an expert or be learned from causal structure learning. Additionally, the RCA algorithm requires an anomaly detection algorithm, which should provide the exact set of observed variables with anomalies \mathcal{V}_A , assuming that $\mathcal{V}_M \supseteq \mathcal{V}_A$ holds and that observation data for all \mathcal{V}_M is provided.

The preprocessing step first creates a set of potential affected variables $\mathcal{V}_G(r)$ for each root cause $r \in \mathcal{V}_R$ using the given causal graph G . It computes all descendant variables of a given root cause r . This computations step has a complexity of $\mathcal{O}(|\mathcal{V}_R|(|\mathcal{V}| + |\mathcal{E}|))$, since it needs to be computed if a path exists from a root cause to variables belonging to the set of observed variables.

$$\mathcal{V}_G(r) := \{m \in \mathcal{V}_M \mid \exists r \in \mathcal{V}_R, r \rightarrow^* m\}$$

Of these variable sets, we only require the observed variables $\mathcal{V}_M \cap \mathcal{V}_G(r)$ since only in those variables, anomalous effects of the root causes may become visible. The preprocessing results, may be computed in advance and can be stored for a fast computation in the next step.

In the actual matching phase, we compute the Jaccard Root Cause Score (JRCS) for all potential root causes $r \in \mathcal{V}_R$.

$$\text{JRCS}(r) := \frac{|\mathcal{V}_A \cap \mathcal{V}_G(r)|}{|\mathcal{V}_A \cup \mathcal{V}_G(r)|} \quad (6.1)$$

The JRCS rates a root cause as more likely if the set of potentially affected observable variables overlaps the most with the set of detected variables containing anomalies. If the sets are identical, the JRCS will score it as equal to one. Figure 6.2 gives a visual example of the JRCS calculation.

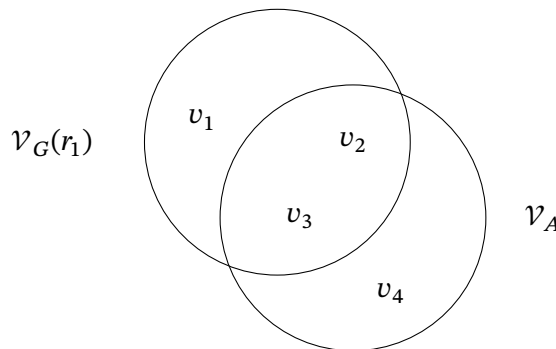


Figure 6.2: Given the shown example sets $\mathcal{V}_G(r_1)$ and \mathcal{V}_A for root cause r_1 , the similarity between the variable sets are scored as $\text{JRCS}(r_1) = 0.5$

6.3 Algorithm Extension with Coarse Temporal Information

In the following, we briefly describe how we adapted the prior introduced algorithm, to allow the integration of coarse temporal knowledge.

For the algorithm to be applied to time series data in industrial applications, we made further adjustments.

To perform the algorithm, we split the process into process steps $p \in \mathcal{P}$, given $\mathcal{P} = \{1, 2, \dots, n\}$. Also, we require the order of assignment of process steps p to correspond to the temporal order.

A causal directed graph $G = (\mathcal{V}, \mathcal{E})$ is required with annotated root causes $\mathcal{V}_R \subset \mathcal{V}$ and known observables $\mathcal{V}_M \subset \mathcal{V}$. It needs annotations of process steps $\mathcal{P}_{a \rightarrow b} = \{p \in \mathbb{N}\}$ to be provided for all $a, b \in \mathcal{V}$ if $a \neq b$ and $a \rightarrow b$ holds. To create $\mathcal{P}_{a \rightarrow b}$ manually, one should list all process steps in which an anomaly in a would show effects in b .

As the anomaly detection should be performed in advance and is not part of the contribution, we require it to provide a tuple set $\mathcal{T}_A = \{(m, p), \dots\}$ given $m \in \mathcal{V}_M$, which contains pairs of the discovered anomalies and the process steps they occurred in.

Accordingly, the graph preprocessing step needs adaptation to provide these tuples. We require a set of process steps $\mathcal{P}_{a \rightarrow b} = \{p \in \mathbb{N}\}$ to be provided for all $a, b \in \mathcal{V}$ if $a \neq b$ and $a \rightarrow b$ holds. To create $\mathcal{P}_{a \rightarrow b}$ for an edge $a \rightarrow b$ manually, one should list all process steps in which an anomaly in a would show effects in b .

Given these set annotations, we can calculate $\mathcal{P}_{r \rightarrow^* m}$ for each path $r \rightarrow^* m$. It represents the maximum possible set of process steps in which $r \in \mathcal{V}_R$ could cause anomalies in $m \in \mathcal{V}_M$ for the specific path. The algorithm to calculate a unique set of $\mathcal{P}_{\{r \rightarrow^* m\}}$ is shown in Algorithm 6.

Since multiple paths $r \rightarrow^* m$ may be possible, we express with $\mathcal{P}_{\{r \rightarrow^* m\}}$ the maximum possible set of process steps for a set of paths. To exclude double entries, we use the *unique* function.

Algorithm 6: Compute valid process steps for path sets

Input : Set of edges $\{r \rightarrow^* m\}_G$ from r to m in graph G ,
 Set of annotated process steps $\mathcal{P}_{a \rightarrow b}$ for any edge

$a \rightarrow b$

Output: Set of process steps $\mathcal{P}_{\{r \rightarrow^* m\}}$ for each path $r \rightarrow^* m$

```

1  $\mathcal{P}_{\{r \rightarrow^* m\}} \leftarrow \emptyset$ 
2 for  $r \rightarrow^* m \in \{r \rightarrow^* m\}_G$  do
3    $i \leftarrow 0$ 
4    $j \leftarrow 1$ 
5   while  $j \leq \text{length}(r \rightarrow^* m)$  do
6      $(a \rightarrow b) \leftarrow (r \rightarrow^* m)[j]$ 
7      $\mathcal{P}_{\text{temp}} \leftarrow \{\mathcal{P}_{a \rightarrow b} \geq i\}$ 
8      $i \leftarrow \min(\mathcal{P}_{\text{temp}})$ 
9      $j \leftarrow j + 1$ 
10   $\mathcal{P}_{\{r \rightarrow^* m\}} \leftarrow \mathcal{P}_{\text{temp}} \cup \mathcal{P}_{\{r \rightarrow^* m\}}$ 
11 return  $\text{unique}(\mathcal{P}_{\{r \rightarrow^* m\}})$ 

```

The algorithm only allows the spread of anomaly information to variables of the same or later occurring process steps. This and the edge annotations with process steps are necessary as the bare summary causal graph does naturally not contain any temporal information about its represented causal relations. After its computation for each potential root cause $r \in \mathcal{V}_R$, each tuple set $\mathcal{T}_G(r)$ can be generated by calculating tuples (m, p) given $m \in \mathcal{V}_M$ and $p \in \mathcal{P}_{\{r \rightarrow^* m\}}$ for each path $r \rightarrow^* m$. Here, $\{r \rightarrow^* m\}_G$ denotes the set of all possible paths from r to m in G .

$$\mathcal{T}_G(r) = \{(m, p) \mid \forall r \rightarrow^* m \in \{r \rightarrow^* m\}_G, p \in \mathcal{P}_{\{r \rightarrow^* m\}}\} \quad (6.2)$$

Since the computation of all possible paths between the root cause and the measurement variables is required, this step has a complexity of $\mathcal{O}(|\mathcal{V}_M| \cdot |\mathcal{V}_R| \cdot |\mathcal{V}|!)$ in the worst case given the underlying graph is acyclic.

After the computation, we can now create the adapted JRCS scores for each root cause r given the tuple sets $\mathcal{T}_G(r)$ and \mathcal{T}_A containing the tuples of individual process steps and individual measurement variables.

$$\text{JRCS}(r) = \frac{|\mathcal{T}_A \cap \mathcal{T}_G(r)|}{|\mathcal{T}_A \cup \mathcal{T}_G(r)|} \quad (6.3)$$

As before, the higher the JRCS result, the better the inspected root cause $r \in \mathcal{V}_R$ fits the anomalous observation \mathcal{T}_A .

6.4 Application on Robotic Gripping Scenario

As part of the Wertstromkinematik project funded by the future fields program of the KIT, we investigated the application of the RCA algorithm on a robotic gripping process simulated with CoppeliaSim V4.5.1 using MatLab and SimuLink. We briefly describe the application environment in Section 6.4.1 and give details about the construction of the causal graph in Section 6.4.2. Then, we discuss relevant root causes and the integration of existing root cause information in Section 6.4.3. In Section 6.4.4, we show the design of the experimental setup and the results.

6.4.1 Application Environment

We apply the extended algorithm to a simple robotic gripping process. In it, we identified five essential process steps which are required for a successful process execution. Each is described in the following step by step. Corresponding images are provided in Figure 6.3.

In the first step, a conveyor moves the prepared workpiece into a gripping position. A proximity sensor is located at the gripping position and measures the distance from the sensor to the gripping position using a laser. The distance decreases significantly if the workpiece is present. In the second step, the coupled robots will move from their default position to the gripping position at the conveyor in unison. Then in the third process step, the magnetic gripper attached to the coupling piece of the robot is powered to attract and grip the workpiece. In step four, the robots move the gripped workpiece from the gripping position to the deposit position. Due to the weight of the gripped workpiece, the torque measure by the robots significantly increases during movement. The sensor back at the gripping position does not indicate the presence of the workpiece anymore. Instead, another proximity sensor at the deposit position starts indicating its presence on its arrival. In the fifth and final process step, the robots disengage the magnetic gripper to leave the workpiece in the final deposit position.

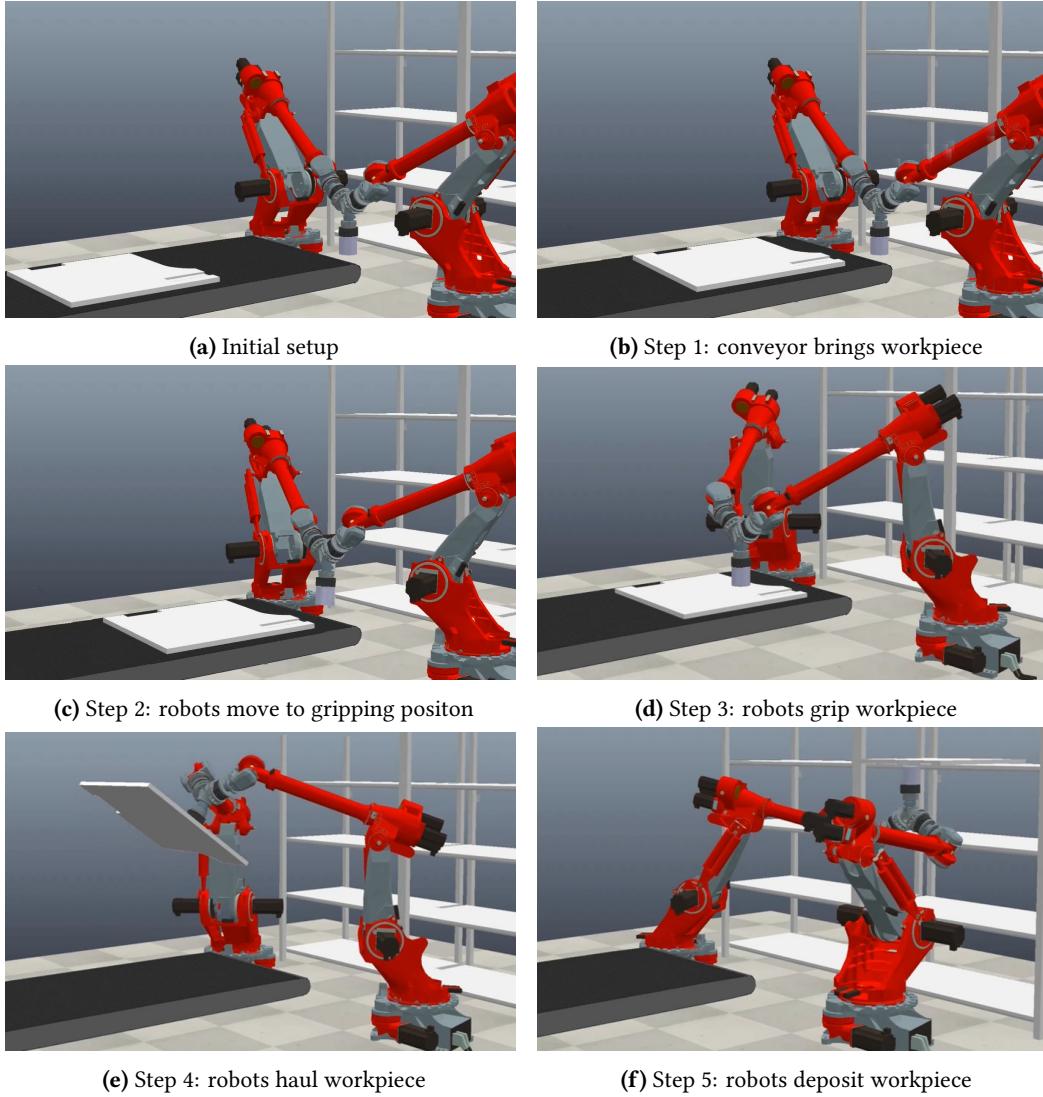


Figure 6.3: Shown are the basic process steps of the gripping simulation. There are proximity sensors located in the shelf and on the belt. The images depict collaborating robots as they were another research topic of the funding project [Müh21, Kim21].

6.4.2 Causal Graph of the Gripping Process

We manually constructed a causal graph for the described scenario and depicted it in Figure 6.4. The corresponding variable sets \mathcal{V}_M and \mathcal{V}_R are indicated by the variables' horizontal placement. The required set of process steps of a causal relation is annotated close to its edge. We simplified the causal graph by summarizing the robots' axis angle and angle speed measurements as one variable each. Commonly, each of the seven robot axes comes with individual measurements. To allow the easy depiction of the graph, some variables with identical names occur several times.

When creating the causal graph keep in mind, that the operating system of industrial robots is commonly a black box with high complexity and such makes the creation of a detailed causal graph for the robots themselves challenging.

We constructed the causal graph with the help of three domain experts from Fraunhofer IOSB and wbk. We interviewed them about the causal relation in the gripping process.

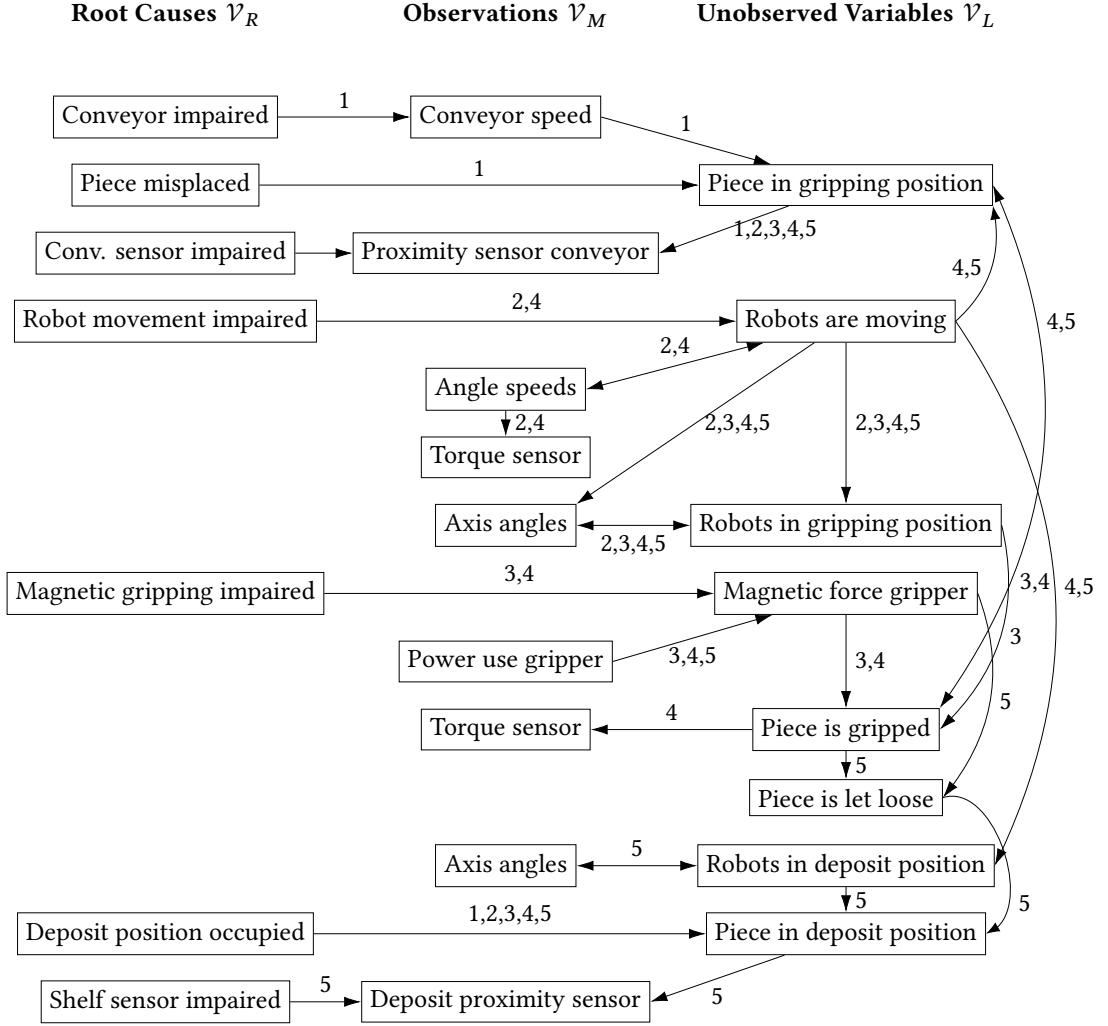


Figure 6.4: The summary causal graph of the robotic setup. The horizontal orientation of the variables indicates their affiliation with the variable sets shown above. Observed values appearing several times are identical. Required sets of process steps of a causal relation are annotated to the edges of the graph.

6.4.3 Inspected Root Causes

The before-described scenario pictures the gripping process in its best case. In practice, it may suffer from various flaws and deficiencies. During the interviews with the domain experts, we also inquired about potential shortcomings in such a gripping process. The results of the investigation is depicted in Table 6.1. The detailed list of potential root causes predicted by the experts are listed in the middle row. We do not claim it to be complete. Since the available information about the process is limited and thus most of the root causes cannot be identified in detail, we chose a higher level of abstraction of root causes (shown in the left column) to practice the algorithms on. We grouped the detailed root causes in Table 6.1 accordingly. We learned that the Operating Systems (OPs) of the conveyor and the robots may provide information for some

root causes. However, we came to the conclusion that without expert assistance or additional designed RCA procedures most root causes cannot be identified per se.

We simulated a fault-free scenario and two additional faulty scenarios and took the indicated observations. One root cause is a shifted workpiece on the conveyor belt that does not reach the gripping position, therefore the robots cannot grab it and also not transport it into the deposit position. The other root cause is a workpiece in the gripping position, where the magnetic force of the gripper is not sufficient to grab it and thus the robots move without the piece into the deposit position.

6.4.4 Experimental Setup and Results

We applied the algorithm described in Section 6.3 on the two anomalous datasets described in Section 6.4.3. The graph in Figure 6.4 was preprocessed as described given the root cause categories from Table 6.1. The results are depicted in Table 6.2.

Table 6.2: Overview of the potential anomalous observations affected by the root causes in the shown process steps. Each row corresponds to $\mathcal{T}_G(r)$ for its respective root cause.

		Potential Anom. Observations \mathcal{V}_M						
Pot. Root Causes \mathcal{V}_R		Conveyor Speed	Prox. Sensor Conveyor	Angle Speeds	Torque Sensor	Axis Angles	Power Use Gripper	Deposit Prox. Sensor
	Conveyor impaired	1	1,2,3,4,5	-	4	-	-	5
	Piece misplaced	-	1,2,3,4,5	-	4	-	-	5
	Conv. Prox. Sensor impaired	-	1,2,3,4,5	-	-	-	-	-
	Robot mov. impaired	-	4,5	2,4	2,4	2,3,4,5	-	5
	Magnetic gripping impaired	-	4,5	-	4	-	-	5
	Shelf Prox. Sensor impaired	-	-	-	-	-	-	1,2,3,4,5
	Deposit position occupied	-	-	-	-	-	-	1,2,3,4,5

For anomaly detection, we used a supervised version of the nearest neighbor algorithm. The distance to the nearest neighbor was used for anomaly scoring. Within each process step, it calculated the Euclidean distance for each inspected time step to the features of its closest normal time step. A sum was formed over all such distances of a process step and divided by the number of the process step's entries. A fixed threshold decided if the inspected process step was detected as normal or anomalous. The set \mathcal{T}_A was formed by retrieving from each process step the observations, which contributed to the Euclidean distance. Table 6.3 shows the anomalous process steps and the corresponding observed variables we discovered as anomalous.

Table 6.1: Overview of potential root causes and their categories. Also, it is indicated by whom the prior knowledge about the root cause may be provided as stated by the interviewed domain experts

Root causes used for algorithms	Process steps	Potential root causes according to experts	Root cause may be know by
Piece out of position on conveyor	1	No piece provided The piece shifted on the conveyor	- -
Conveyor impaired	1	Conveyor jammed Motor broken Manual emergency stop Process halted manually No power supply Motor broken	conveyor OS - conveyor OS conveyor OS conveyor OS conveyor OS
Robot movement impaired	2, 4	Positioning uncalibrated Rotor wear Computation error Sensor malfunction Breaks are not deactivated Collision with internal boundary Collision with object Manual emergency stop Process halted manually No power supply Gripping position out of reach Person in area of action Deposit position out of reach	- - - - robot OS robot OS robot OS robot OS robot OS robot OS robot OS robot OS robot OS robot OS
Magnetic gripping impaired	3, 4	Material flaw Magnetic force too low Wrong material Workpiece malformed Movement torque too high	- - - - -
Proximity sensor impaired	1, 5	No workpiece provided Sensor malfunction	- -
Deposit position out of order	1,2,3,4,5	Other piece already in shelf	-

Table 6.3: Overview of the discovered anomalous observations and their corresponding process steps. Each row corresponds to \mathcal{T}_A for its respective scenario.

	Detected Anomalous Measurements \mathcal{V}_M						
	Conveyor Speed	Prox. Sensor Conveyor	Angle Speeds	Torque Sensor	Axis Angles	Power Use Gripper	Deposit Prox. Sensor
Faulty Scenario I	-	1,2,3,4,5	-	4	-	-	5
Faulty Scenario II	-	4,5	-	4	-	-	5

Given the detected anomalies (\mathcal{T}_A) shown in Table 6.3 and the potential anomalous observations per root cause ($\mathcal{T}_G(r)$) in Table 6.2, we calculated the JRCS for each potential root cause for both inspected faulty simulation scenarios. In Table 6.4, we show the JRCS result for our new temporal extension and the basic algorithm.

Table 6.4: Jaccard Root Cause Scores for both simulated anomalous scenarios and each root cause for developed approaches. A value of one indicates an ideal match between the root causes and the detected anomalies. We observe better distinction between the root causes, if temporal information is considered.

		Basic Algorithm		Extended Algorithm	
		Scenario I	Scenario II	Scenario I	Scenario II
Pot. Root Causes \mathcal{V}_R	Conveyor impaired	0.875	0.500	0.750	0.750
	Piece misplaced	1	0.571	1	1
	Conv. prox. sensor impaired	0.625	0.286	0.333	0.333
	Robot mov. impaired	0.286	0.364	0.600	0.600
	Magnetic gripping impaired	0.400	1	1	1
	Shelf prox. sensor impaired	0.111	0.125	0.333	0.333
	Deposit position occupied	0.111	0.125	0.333	0.333

We can observe that the original algorithm, which considers only the variables to compute the JRCS, cannot distinguish between two root causes because they affect the same set of variables. In comparison, the extended algorithm rates the actual root causes as the likeliest. The temporal algorithm scores other root causes, which affect similar observed variables and process steps as depicted in Table 6.2, as likelier. Root causes with identical potentially affected variables $\mathcal{T}_G(r)$, like the last two root cause entries in the table, are rated as equally likely. This demonstrates that the algorithm does not guarantee a clear distinction between root causes. This may be resolved by adding sensors to the process, which monitor the different effects of these root causes.

6.5 Discussion

The two approaches to identify potential root causes for given sets of anomalies are novel in their use of causal graphs to draw deductions about potential root causes. They both primarily rely on a causal graph with annotations to be provided. On the one hand, this makes both approaches highly dependent on the availability and quality of prior knowledge. On the other hand, this may contribute to the conservation and application of expert knowledge. To decrease the amount of required prior knowledge, we purposefully chose the use of more compact summary graphs over the far larger timeseries graphs.

When the causal graphs are preprocessed, we potentially overapproximate the affected observable variables since we assume all observable variables dependent on a root cause variable to be potentially affected by it. It is not considered yet, if the anomalous causal information can travel as far, and if the resulting anomaly in the observed variable would still be detectable or of significance. In theory, if the causal graph is cyclic, as may be relevant for example of cyclic production processes, the set of potentially affected observations may become very large and impact the quality of the root cause matching. Also, both approaches have the flaw to consider few information about the detected anomalies: while one considers only the variables of occurrence, the other considers additionally coarse temporal information. This entails a weakness to differentiate between anomalies present in the same variables at similar times, but also leaves room for future improvements.

We argue that both developed approaches are naturally explainable, because we deem causal graphs as well as the algorithm's preprocessing and matching step to be intuitive for human understanding. From our experience, humans intuitively investigate potential root causes by comparing and arguing about their potential effects. Also, they assess the potential root causes according to their similarity to the detected anomaly patterns and consider the best possible fit to be the most likely. This way, the approaches may extend human understanding by allowing the fast inference on the provided causal graphs.

We want to mention, that the overall design is inspired by the design of recommender systems. As is typically for them, the more computation heavy preprocessing step may be computed in advance offline, while the actual matching of the prepared information to a query (the detected anomalies) happens in an efficient calculation online using for example the Jaccard similarity [Bag19].

6.6 Conclusion and Outlook

In this chapter, we present a root cause algorithm that considers a given causal graph to infer from anomalous observations a recommendation on the specific unobserved root cause. It requires an anomaly detection algorithm and a given timeseries summary causal graph annotated with root causes and observable variables to match the discovered anomalies to potential root causes. The anomaly detection step itself is not part of our contribution. The algorithm preprocesses in the first step the provided causal graph to identify potential observed variables affected by each root

cause. Then, during the matching phase, the Jaccard Root Cause Score (JRCS) is used to match prior discovered anomalies in the observed variables to the potential root causes based on the fit in observed variables.

We also introduce an adaptation of the algorithm which uses coarse grained temporally information annotated in the causal graph to draw temporal valid conclusions about potential root causes. This is based on the assumption that anomaly information is essential causal information which can only spread to observations later in time. The usefulness of the developed approaches is demonstrated on a robotic gripper process.

Future work might include the additional consideration of the specific form of anomalies. Several root causes might affect sensor measurements in the same variables at the same process steps but in different manners.

7 Conclusion of the Thesis

In this thesis, we contributed to the domain of causal structure learning by achieving progress in the areas of structure learning, structure evaluation and the use of causal structures for root cause analysis.

Intervention-driven experiments are a main capability of human causal structure learning and provide the possibility to test causal knowledge in vivo. Established interventions in causal research primarily inspect probability distributions to identify underlying causal relations. They may inflict hard changes on the investigated system, or may require large quantities of data to reliably detect the effects.

We proposed new intervention methods based on the injection and recovery of signals to learn the underlying causal relations. The interventions are low-invasive, as they only inflict a minor change on the variable of injection and thus are more similar to noise. We came up with two different techniques. One creates an identifiable signal by modifying its amplitude to add it to the injection variable and tries to recover it via pattern matching in the observed variables. The other does so by adding a sine-based signal of a chosen frequency to the targeted variable and recovering it in the observed variables by checking for the frequency of injection using the Fast Fourier transform. As the signals travel with the underlying causal relations, their rediscovery discloses information about present causal paths. If additional information is collected about the temporal occurrence, we can gain additional constraints due to the temporal law of causality. Both approaches have shown to perform well, but not fully reliable, in our experiments. In their results, the frequency modulation-based approach performed slightly better than the amplitude modulation-based approach. Future work may be dedicated to real-life application of the developed approaches.

We developed novel representations of equivalence classes, which play a main role in constraint-based structure learning. They efficiently keep track of the learned evidence for present and absent causal relations and the corresponding causal graph candidates. For this purpose, we have incorporated established principles as Binary Decision Diagrams and the Kleene Ternary Logic into the causal structure learning process. Additionally, we have enhanced the existing PC structure learning algorithm to use the novel principles and also detect absent causal relations more efficiently. We showed in example applications, that it may require less time and fewer conditional independence tests to deliver identical results. In future work, one may contribute to the use of the representations in current structure learning algorithms. This may for example include the PC-MCI algorithm [Run19] for timeseries data, which is a descendant of the PC algorithm.

We performed an inspection of current scoring methods for the evaluation of discovered causal graphs using ground truth. The use of such scoring methods, like the F_1 score, the TPR, the PRC

AUC and the ROC AUC as well as the SHD, is standard in causal structure learning, but while learning methods steadily improve in discovery power and number, few works were dedicated to scoring methods for causal structure learning. By now, the types of learned causal structures are as diverse as the different types of causal structure learning methods themselves. This way, a general benchmark is not possible.

We developed several criteria for structure learning methods to deliver reliable evaluations of the discovered causal graphs and inspected these criteria for current scoring methods. Part of the evaluation is our novel scoring method called normalized Causal Edit Distance (nCED). By the use of a universal notation, it may evaluate discovered DAGs and CPDAGs. As is novel, it also performs on MAGs and PAGs and may take besides a DAG also a MAG as ground truth for evaluation. This opens up the potential for broader benchmarks of methods that learn different graph types. In doing so, the scoring method also allows flexibility in the scoring of undirected edges in comparison to absent or present edges.

The scoring method allows further development as its universal representation generalizes some special peculiarities of the different causal graphs to make them comparable. For example, it evaluates bidirected edges of PAGs not as an indication of a latent confounder, but as its common representation as a bidirected edge.

A large quantity of algorithms exists which can detect anomalies in multivariate data in machine learning, however to treat anomalies and their root cause, we need competent algorithms that can use them to deduce their root cause. For this purpose, we developed a new approach, which uses a given, annotated causal structure and its ingrained ability to draw deductions about the spread of (anomalous) causal information in the represented system. The deductions became possible by a new scoring method called the Jaccard Root Cause Score, which recommends root causes according to their fit in potentially affected observables to the given discovered anomalies and, if desired, also by the fit in the temporal spread of anomaly information from the root cause to the affected observable in the observed system.

The developed approach requires a given causal graph and may require temporal information of anomalies annotated by an expert. This leads to a high demand for prior knowledge for the algorithm to work. On the other hand, it has the advantage that the causal graph can preserve a considerable amount of expert knowledge and make it accessible in the search for the root cause.

Future work may focus on the integration of additional criteria into the root cause search, since currently it only considers the anomaly indicating variables in general, the coarse temporal information about the presence of the variable and the temporal spread in the causal graph. Also, an investigation into using the developed approach together with structure learning algorithms may contribute to its capabilities. The underlying graph would not need to be created by an expert by hand, instead, the graph may be flexibly relearned if it becomes out of date in a dynamic process.

Bibliography

- [Abu15] ABU-AISHEH, Zeina; RAVEAUX, Romain; RAMEL, Jean-Yves and MARTINEAU, Patrick: “An exact graph edit distance algorithm for solving pattern recognition problems”. In: *4th International Conference on Pattern Recognition Applications and Methods 2015*. 2015.
- [Aci03] ACID, Silvia and CAMPOS, Luis M de: “Searching for Bayesian network structures in the space of restricted acyclic partially directed graphs”. In: *Journal of Artificial Intelligence Research* 18 (2003), pp. 445–490.
- [Agr16] AGRAWAL, Vedika; PANIGRAHI, Bijaya Ketan and SUBBARAO, PMV: “Intelligent decision support system for detection and root cause analysis of faults in coal mills”. In: *IEEE Transactions on Fuzzy Systems* 25.4 (2016), pp. 934–944.
- [Aka98] AKAIKE, Hirotugu: “Information theory and an extension of the maximum likelihood principle”. In: *Selected papers of hirotugu akaike*. Springer, 1998, pp. 199–213.
- [Ake78] AKERS: “Binary decision diagrams”. In: *IEEE Transactions on computers* 100.6 (1978), pp. 509–516.
- [And13] ANDERSEN, Holly: “When to expect violations of causal faithfulness and why it matters”. In: *Philosophy of Science* 80.5 (2013), pp. 672–683.
- [And22] ANDERSON, R Bruce and GERAS, Matthew: “Correlation Versus Causation”. In: *Encyclopedia of Big Data*. Springer, 2022, pp. 225–228.
- [Ass22a] ASSAAD, Charles K; DEVIJVER, Emilie and GAUSSIER, Eric: “Entropy-based discovery of summary causal graphs in time series”. In: *Entropy* 24.8 (2022), p. 1156.
- [Ass22b] ASSAAD, Charles K; DEVIJVER, Emilie and GAUSSIER, Eric: “Survey and evaluation of causal discovery methods for time series”. In: *Journal of Artificial Intelligence Research* 73 (2022), pp. 767–819.
- [Ass23] ASSAAD, Charles K; EZ-ZEJJARI, Imad and ZAN, Lei: “Root Cause Identification for Collective Anomalies in Time Series given an Acyclic Summary Causal Graph with Loops”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 8395–8404.
- [Bag19] BAG, Sujoy; KUMAR, Sri Krishna and TIWARI, Manoj Kumar: “An efficient recommendation generation using relevant Jaccard similarity”. In: *Information Sciences* 483 (2019), pp. 53–64.
- [Bee81] BEEBEE, Helen: *Hume and the Problem of Causation*. Oxford Handbooks, 1981.
- [Bie09] BIERE, Armin; HEULE, Marijn and MAAREN, Hans van: *Handbook of satisfiability*. IOS press, 2009.

- [Bor12] BORBOUDAKIS, Giorgos and TSAMARDINOS, Ioannis: “Incorporating Causal Prior Knowledge as Path-Constraints in Bayesian Networks and Maximal Ancestral Graphs”. In: *Proceedings of the 29th ICML*. Edinburgh, Scotland, 2012, pp. 427–434.
- [Bro22] BROUILLARD, Philippe; TASLAKIAN, Perouz; LACOSTE, Alexandre; LACHAPELLE, Sébastien and DROUIN, Alexandre: “Typing assumptions improve identification in causal discovery”. In: *Conference on Causal Learning and Reasoning*. PMLR. 2022, pp. 162–177.
- [Bry92] BRYANT, Randal E: “Symbolic boolean manipulation with ordered binary-decision diagrams”. In: *ACM Computing Surveys (CSUR)* 24.3 (1992), pp. 293–318.
- [Cam07] CAMPBELL, John: “An interventionist approach to causation in psychology”. In: *Causal learning: Psychology, philosophy, and computation* (2007), pp. 58–66.
- [Cha09] CHANDOLA, Varun; BANERJEE, Arindam and KUMAR, Vipin: “Anomaly detection: A survey”. In: *ACM computing surveys (CSUR)* 41.3 (2009), pp. 1–58.
- [Cha90] CHANG, Chung Chien and YU, Cheng Ching: “On-line fault diagnosis using the signed directed graph”. In: *Industrial & Engineering Chemistry Research* 29.7 (1990), pp. 1290–1299.
- [Che22] CHENG, Lu; GUO, Ruocheng; MORAFFAH, Raha; SHETH, Paras; CANDAN, K Selçuk and LIU, Huan: “Evaluation methods and measures for causal learning algorithms”. In: *IEEE Transactions on Artificial Intelligence* 3.6 (2022), pp. 924–943.
- [Chi96] CHICKERING, David Maxwell: “Learning Bayesian networks is NP-complete”. In: *Learning from data: Artificial intelligence and statistics V* (1996), pp. 121–130.
- [Con19] CONSTANTINOU, Anthony C: “Evaluating structure learning algorithms with a balanced scoring function”. In: *arXiv preprint arXiv:1905.12666* (2019).
- [Con21] CONSTANTINOU, Anthony C; FENTON, Norman and NEIL, Martin: “How do some Bayesian Network machine learned graphs compare to causal knowledge?” In: *arXiv preprint arXiv:2101.10461* (2021).
- [Coo65] COOLEY, James W and TUKEY, John W: “An algorithm for the machine calculation of complex Fourier series”. In: *Mathematics of computation* 19.90 (1965), pp. 297–301.
- [DAm22] D’AMOUR, Alexander; HELLER, Katherine; MOLDOVAN, Dan; ADLAM, Ben; ALIPANAHI, Babak; BEUTEL, Alex; CHEN, Christina; DEATON, Jonathan; EISENSTEIN, Jacob; HOFFMAN, Matthew D et al.: “Underspecification presents challenges for credibility in modern machine learning”. In: *The Journal of Machine Learning Research* 23.1 (2022), pp. 10237–10297.
- [Dau92] DAUBECHIES, Ingrid: Ten lectures on wavelets. SIAM, 1992.
- [Dav06] DAVIS, Jesse and GOADRICH, Mark: “The relationship between Precision-Recall and ROC curves”. In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 233–240.

- [Die22] DIEDRICH, Alexander; BUCHHOLZ, Franziska and NIGGEMANN, Oliver: “Learning Causal System Descriptions for Diagnosing Cyber-Physical Systems”. In: *33rd International Workshop on Principle of Diagnosis–DX 2022*. 2022.
- [Ebe14] EBERHARDT, Frederick: “Direct causes and the trouble with soft interventions”. In: *Erkenntnis* 79 (2014), pp. 755–777.
- [Ebe07] EBERHARDT, Frederick and SCHEINES, Richard: “Interventions and causal inference”. In: *Philosophy of science* 74.5 (2007), pp. 981–995.
- [Eme23] EMEZUE, Chris Chinenye; DROUIN, Alexandre; DELEU, Tristan; BAUER, Stefan and BENGIO, Yoshua: “Benchmarking Bayesian Causal Discovery Methods for Downstream Treatment Effect Estimation”. In: *ICML 2023 Workshop on Structured Probabilistic Inference and Generative Modeling*. 2023.
- [Faw04] FAWCETT, Tom: “ROC graphs: Notes and practical considerations for researchers”. In: *Machine learning* 31.1 (2004), pp. 1–38.
- [Fis15] FISHER, Ronald A: “Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population”. In: *Biometrika* 10.4 (1915), pp. 507–521.
- [Fis36] FISHER, Ronald Aylmer: “Design of experiments”. In: *British Medical Journal* 1.3923 (1936), p. 554.
- [Fou88] FOURIER, Jean Baptiste Joseph: *Théorie analytique de la chaleur*. Vol. 1. Gauthier-Villars, 1888.
- [Gao10] GAO, Xinbo; XIAO, Bing; TAO, Dacheng and LI, Xuelong: “A survey of graph edit distance”. In: *Pattern Analysis and applications* 13 (2010), pp. 113–129.
- [Gar97] GARCIA, E Alcorta and FRANK, Paul M: “Deterministic nonlinear observer-based approaches to fault diagnosis: A survey”. In: *control engineering practice* 5.5 (1997), pp. 663–670.
- [Gly19] GLYMOUR, Clark; ZHANG, Kun and SPIRITES, Peter: “Review of causal discovery methods based on graphical models”. In: *Frontiers in genetics* 10 (2019), p. 524.
- [Gra69] GRANGER, Clive WJ: “Investigating causal relations by econometric models and cross-spectral methods”. In: *Econometrica: journal of the Econometric Society* (1969), pp. 424–438. DOI: [10.2307/1912791](https://doi.org/10.2307/1912791).
- [Han23] HAN, Xiao; ZHANG, Lu; WU, Yongkai and YUAN, Shuhan: “On Root Cause Localization and Anomaly Mitigation through Causal Inference”. In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2023, pp. 699–708.
- [Hau12] HAUSER, Alain and BÜHLMANN, Peter: “Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs”. In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 2409–2464.
- [Hec95] HECKERMAN, David; GEIGER, Dan and CHICKERING, David M: “Learning Bayesian networks: The combination of knowledge and statistical data”. In: *Machine learning* 20 (1995), pp. 197–243.

- [Hla07] HLAVÁČKOVÁ-SCHINDLER, Katerina; PALUŠ, Milan; VEJMEKKA, Martin and BHATTACHARYA, Joydeep: “Causality detection based on information-theoretic approaches in time series analysis”. In: *Physics Reports* 441.1 (2007), pp. 1–46.
- [Hue20] HUEGLE, Johannes; HAGEDORN, Christopher and UFLACKER, Matthias: “How Causal Structural Knowledge Adds Decision-Support in Monitoring of Automotive Body Shop Assembly Lines.” In: *IJCAI*. 2020, pp. 5246–5248.
- [Hyt13] HYTTINEN, Antti; HOYER, Patrik O; EBERHARDT, Frederick and JARVISALO, Matti: “Discovering cyclic causal models with latent variables: A general SAT-based procedure”. In: *Proceeding of UAI 2013* (2013).
- [Ili14] ILIN, Vladimir; STEVENSON, Ian H and VOLGUSHEV, Maxim: “Injection of fully-defined signal mixtures: a novel high-throughput tool to study neuronal encoding and computations”. In: *PLoS One* 9.10 (2014), e109928.
- [Iri79] IRI, Masao; AOKI, Katsuaki; O’SHIMA, Eiji and MATSUYAMA, H: “An algorithm for diagnosis of system failures in the chemical process”. In: *Computers & Chemical Engineering* 3.1-4 (1979), pp. 489–493.
- [Ise97] ISERMANN, Rolf and BALLE, Peter: “Trends in the application of model-based fault detection and diagnosis of technical processes”. In: *Control engineering practice* 5.5 (1997), pp. 709–719.
- [JIA02] JIA, RONG-QING: “Cascade algorithms in wavelet analysis”. In: *Wavelet Analysis: Twenty Years’ Developments* (2002), pp. 196–230.
- [Jon09] JONGH, Martijn de and DRUZDZEL, Marek J: “A comparison of structural distance measures for causal Bayesian network models”. In: *Recent Advances in Intelligent Information Systems, Challenging Problems of Science, Computer Science series* (2009), pp. 443–456. URL: <http://d-scholarship.pitt.edu/6011/>.
- [Kim21] KIMMIG, Andreas et al.: “Wertstromkinematik–Produktionssysteme neu gedacht”. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 116.12 (2021), pp. 935–939.
- [Kle38] KLEENE, Stephen Cole: “On notation for ordinal numbers”. In: *The Journal of Symbolic Logic* 3.4 (1938), pp. 150–155.
- [Kol09] KOLLER, Daphne and FRIEDMAN, Nir: Probabilistic graphical models: principles and techniques. MIT press, 2009. DOI: [10.5555/1795555](https://doi.org/10.5555/1795555).
- [Kor04] KORB, Kevin B; HOPE, Lucas R; NICHOLSON, Ann E and AXNICK, Karl: “Varieties of causal intervention”. In: *PRICAI 2004: Trends in Artificial Intelligence: 8th Pacific Rim International Conference on Artificial Intelligence. Proceedings* 8. Springer. 2004, pp. 322–331.
- [Kou22] KOUTROULIS, Georgios; MUTLU, Belgin and KERN, Roman: “A causality-inspired approach for anomaly detection in a water treatment testbed”. In: *Sensors* 23.1 (2022), p. 257.

- [Kra23] KRAUSS, Rune; GOLI, Mehran and DRECHSLER, Rolf: “EDDY: A multi-core BDD package with dynamic memory management and reduced fragmentation”. In: *Proceedings of the 28th Asia and South Pacific Design Automation Conference*. 2023, pp. 423–428.
- [Küh13] KÜHNERT, Christian: Data-driven methods for fault localization in process technology. Vol. 15. KIT Scientific Publishing, 2013.
- [Küh11] KÜHNERT, Christian; BERNARD, Thomas and FREY, Christian: “Causal structure learning in process engineering using Bayes Nets and soft interventions”. In: *2011 9th IEEE International Conference on Industrial Informatics*. IEEE. 2011, pp. 69–74.
- [Küh19] KÜHNERT, Christian; FREY, Christian and SEYBOLDT, Ruben: “Detection of Directed Connectivities in Dynamic Systems for Different Excitation Signals using Spectral Granger Causality”. In: *Machine Learning for Cyber Physical Systems*. Ed. by BEYERER, Jürgen; KÜHNERT, Christian and NIGGEMANN, Oliver. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 97–106.
- [Lau88] LAURITZEN, Steffen L and SPIEGELHALTER, David J: “Local computations with probabilities on graphical structures and their application to expert systems”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 50.2 (1988), pp. 157–194.
- [Law20] LAWRENCE, Andrew R.; KAISER, Marcus; SAMPAIO, Rui and SIPOS, Maksim: “Data Generating Process to Evaluate Causal Discovery Techniques for Time Series Data”. In: *Causal Discovery and Causality-Inspired Machine Learning Workshop at Neural Information Processing Systems* (2020).
- [Lju05] LJUNG, Lennart: “System identification”. In: *Uncertainty and Control: Proceedings of an International Seminar Organized by Deutsche Forschungs-und Versuchsanstalt für Luft-und Raumfahrt (DFVLR) Bonn, Germany, May, 1985*. Springer. 2005, pp. 48–83.
- [Lon10] LONDON, Michael; ROTH, Arnd; BEEREN, Lisa; HÄUSSER, Michael and LATHAM, Peter E: “Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex”. In: *Nature* 466.7302 (2010), pp. 123–127.
- [Mag16] MAGLIACANE, Sara; CLAASSEN, Tom and MOOIJ, Joris M: “Joint causal inference on observational and experimental datasets”. In: *arXiv preprint arXiv:1611.10351* (2016).
- [Mee95] MEEK, Christopher: “Causal Inference and Causal Explanation with Background Knowledge”. In: *Proceedings of UAI 1995*. 1995, pp. 403–410.
- [Men22] MENEGOZZO, Giovanni; DALL’ALBA, Diego and FIORINI, Paolo: “CIPCaD-Bench: continuous industrial process datasets for benchmarking causal discovery methods”. In: *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2022, pp. 2124–2131.
- [Moo20] MOOIJ, Joris M; MAGLIACANE, Sara and CLAASSEN, Tom: “Joint causal inference from multiple contexts”. In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 3919–4026.

- [Mue22] MUEEN, Abdullah; ZHING, Sheng; ZHU, Yan; YEH, Michael; KAMGAR, Kaveh; VISWANATHAN, Krishnamurthy; GUPTA, Chetan and KEOGH, Eamonn: The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance. <http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html>. Aug. 2022.
- [Müh21] MÜHLBEIER, Edgar; OEXLE, Florian; GÖNNHEIMER, Philipp and FLEISCHER, Jürgen: “Wertstromkinematik–Produktionssysteme neu gedacht”. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 116.11 (2021), pp. 847–851.
- [ODo06] O’DONNELL, RT; NICHOLSON, Ann E; HAN, B; KORB, Kevin B; ALAM, MJ and HOPE, LR: “Incorporating expert elicited structural information in the CaMML causal discovery program”. In: *Proceedings of the 19th AJCAI*. 2006, pp. 1–16.
- [Par18] PARIDA, Pramod Kumar; MARWALA, Tshilidzi and CHAKRAVERTY, Snehashish: “A multivariate additive noise model for complete causal discovery”. In: *Neural Networks* 103 (2018), pp. 44–54.
- [Pat91] PATTON, Ron J and CHEN, Jie: “A review of parity space approaches to fault diagnosis”. In: *IFAC Proceedings Volumes* 24.6 (1991), pp. 65–81.
- [Pea00] PEARL, Judea: *Causality: Models, reasoning and inference*. Cambridge University Press, 2000.
- [Pea09] PEARL, Judea: *Causality*. Cambridge university press, 2009.
- [Pea10] PEARL, Judea: “Causal Inference”. In: *Proceedings of Workshop on Causality: Objectives and Assessment at NIPS 2008*. Ed. by GUYON, Isabelle; JANZING, Dominik and SCHÖLKOPF, Bernhard. Vol. 6. Proceedings of Machine Learning Research. Whistler, Canada: PMLR, Dec. 2010, pp. 39–58.
- [Pea12] PEARL, Judea: “The do-calculus revisited”. In: *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*. 2012, pp. 3–11.
- [Pea19] PEARL, Judea: “The seven tools of causal inference, with reflections on machine learning”. In: *Communications of the ACM* 62.3 (2019), pp. 54–60.
- [Pen07] PENN, Derek C and POVINELLI, Daniel J: “Causal cognition in human and nonhuman animals: A comparative, critical review”. In: *Annu. Rev. Psychol.* 58 (2007), pp. 97–118.
- [Pér18] PÉREZ-SUAY, Adrián and CAMPS-VALLS, Gustau: “Causal inference in geoscience and remote sensing from observational data”. In: *IEEE Transactions on Geoscience and Remote Sensing* 57.3 (2018), pp. 1502–1513.
- [Per17] PERKOVIĆ, Emilija; KALISCH, Markus and MAATHUIS, Maloes H: “Interpreting and using CPDAGs with background knowledge”. In: *Proceedings of UAI 2017*. UAI2017. 2017, p. 120.
- [Pet17] PETERS, Jonas; JANZING, Dominik and SCHÖLKOPF, Bernhard: *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.
- [Pet14] PETERS, Jonas; MOOIJ, Joris M; JANZING, Dominik and SCHÖLKOPF, Bernhard: “Causal discovery with continuous additive noise models”. In: (2014).

- [Pet54] PETERSON, WWTG; BIRDSALL, T and FOX, We: “The theory of signal detectability”. In: *Transactions of the IRE professional group on information theory* 4.4 (1954), pp. 171–212. DOI: [10.1109/TIT.1954.1057460](https://doi.org/10.1109/TIT.1954.1057460).
- [Pfo19] PFOHL, Stephen R; DUAN, Tony; DING, Daisy Yi and SHAH, Nigam H: “Counterfactual reasoning for fair clinical risk prediction”. In: *Machine Learning for Healthcare Conference*. PMLR. 2019, pp. 325–358.
- [Pfr24] PFROMMER, Julius; REHAK, Josephine; ARABIZADEH, Negar and BEYERER, Jürgen: “Causal Reasoning with Path-Constrained Equivalence Classes”. In: (2024). submitted.
- [Reh23a] REHAK, Josephine: “Constraint-based Causal Discovery by using Path Constraints gained from Signal Injection and Recovery”. In: *Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. 2023, p. 47.
- [Reh24a] REHAK, Josephine: “Root Cause Analysis via Anomaly Detection and Causal Graphs”. In: *Dagstuhl Reports* 14 (2024). accepted.
- [Reh24b] REHAK, Josephine; FALKENSTEIN, Alexander and BEYERER, Jürgen: “Metrics for the Evaluation of Learned Causal Graphs Based on Ground Truth”. In: *Machine Learning for Cyber Physical Systems: Selected papers from the International Conference ML4CPS 2024*. accepted. 2024.
- [Reh23b] REHAK, Josephine; SOMMER, Anouk; BECKER, Maximilian; PFROMMER, Julius and BEYERER, Jürgen: “Counterfactual Root Cause Analysis via Anomaly Detection and Causal Graphs”. In: *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*. IEEE. 2023, pp. 1–7.
- [Reh24c] REHAK, Josephine; YOUSSEF, Shahenda and BEYERER, Jürgen: “Root Cause Analysis Using Anomaly Detection and Temporal Informed Causal Graphs”. In: *Machine Learning for Cyber Physical Systems: Selected papers from the International Conference ML4CPS 2024*. accepted. 2024.
- [Rei44] REICHENBACH, Hans: *Philosophic Foundations of Quantum Mechanics*. University of California Press, 1944.
- [Rey03] REYES, Alex D: “Synchrony-dependent propagation of firing rate in iteratively constructed networks in vitro”. In: *Nature neuroscience* 6.6 (2003), pp. 593–599.
- [Rib16] RIBEIRO, Marco Tulio; SINGH, Sameer and GUESTRIN, Carlos: “Why should i trust you? Explaining the predictions of any classifier”. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.
- [Ric02] RICHARDSON, Thomas and SPIRTES, Peter: “Ancestral graph Markov models”. In: *The Annals of Statistics* 30.4 (2002), pp. 962–1030.
- [Ric96] RICHARDSON, Thomas S: “Models of feedback: interpretation and discovery”. PhD thesis. Carnegie-Mellon University, 1996.
- [Ric03] RICHARDSON, Thomas S and SPIRTES, Peter: “Causal inference via ancestral graph models”. In: *Oxford Statistical Science Series* (2003), pp. 83–105.

- [Roo04] ROONEY, James J and HEUVEL, Lee N Vanden: “Root cause analysis for beginners”. In: *Quality progress* 37.7 (2004), pp. 45–56.
- [Run12] RUNGE, Jakob; HEITZIG, Jobst; PETOUKHOV, Vladimir and KURTHS, Jürgen: “Escaping the curse of dimensionality in estimating multivariate transfer entropy”. In: *Physical review letters* 108.25 (2012), p. 258701.
- [Run19] RUNGE, Jakob; NOWACK, Peer; KRETSCHMER, Marlene; FLAXMAN, Seth and SEJDINOVIC, Dino: “Detecting and quantifying causal associations in large nonlinear time series datasets”. In: *Science advances* 5.11 (2019), eaau4996.
- [Sac05] SACHS, Karen; PEREZ, Omar; PE’ER, Dana; LAUFFENBURGER, Douglas A and NOLAN, Garry P: “Causal protein-signaling networks derived from multiparameter single-cell data”. In: *Science* 308.5721 (2005), pp. 523–529. DOI: [10.1145/3501714.3501755](https://doi.org/10.1145/3501714.3501755).
- [Sas07] SASAKI, Yutaka et al.: “The truth of the F-measure”. In: *Teach tutor mater* 1.5 (2007), pp. 1–5.
- [Sch22] SCHÖLKOPF, Bernhard: “Causality for machine learning”. In: *Probabilistic and Causal Inference: The Works of Judea Pearl*. 2022, pp. 765–804.
- [Sch97] SCHRICK, Dirk van: “Remarks on terminology in the field of supervision, fault detection and diagnosis”. In: *IFAC Proceedings Volumes* 30.18 (1997), pp. 959–964.
- [Sha18] SHAH, Syed Yousaf; DANG, Xuan-Hong and ZERFOS, Petros: “Root cause detection using dynamic dependency graphs from time series data”. In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 1998–2003.
- [Sma88] SMART, Chris: “Artificial tracer techniques for the determination of the structure of conduit aquifers”. In: *Groundwater* 26.4 (1988), pp. 445–453.
- [Sol17] SOLÉ, Marc; MUNTÉS-MULERO, Victor; RANA, Annie Ibrahim and ESTRADA, Giovanni: “Survey on models and techniques for root-cause analysis”. In: *arXiv preprint arXiv:1701.08546* (2017).
- [Som98] SOMENZI, Fabio: “CUDD: CU decision diagram package release 2.3. 0”. In: *University of Colorado at Boulder* 621 (1998).
- [Spi00] SPIRITES, Peter; GLYMOUR, Clark N; SCHEINES, Richard and HECKERMAN, David: Causation, prediction, and search. MIT press, 2000. DOI: [10.7551/mitpress/1754.001.0001](https://doi.org/10.7551/mitpress/1754.001.0001).
- [Squ23] SQUIRES, Chandler and UHLER, Caroline: “Causal structure learning: A combinatorial perspective”. In: *Foundations of Computational Mathematics* 23.5 (2023), pp. 1781–1815.
- [Tho97] THOMAS, Launey J and WICHMANN, Eyvind H: “On the causal structure of Minkowski spacetime”. In: *Journal of Mathematical Physics* 38.10 (1997), pp. 5044–5086.
- [Tia01] TIAN, Jin and PEARL, Judea: “Causal Discovery from Changes”. In: *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*. 2001, pp. 512–521.

- [Tri15] TRIANTAFILLOU, Sofia and TSAMARDINOS, Ioannis: “Constraint-based causal discovery from multiple interventions over overlapping variable sets”. In: *The Journal of Machine Learning Research* 16.1 (2015), pp. 2147–2205.
- [Tsa06] TSAMARDINOS, Ioannis; BROWN, Laura E and ALIFERIS, Constantin F: “The max-min hill-climbing Bayesian network structure learning algorithm”. In: *Machine learning* 65.1 (2006), pp. 31–78. DOI: [10.1007/s10994-006-6889-7](https://doi.org/10.1007/s10994-006-6889-7).
- [Ume80] UMEDA, T; KURIYAMA, T; O'SHIMA, E and MATSUYAMA, H: “A graphical approach to cause and effect analysis of chemical processing systems”. In: *Chemical Engineering Science* 35.12 (1980), pp. 2379–2388.
- [Ven03a] VENKATASUBRAMANIAN, Venkat; RENGASWAMY, Raghunathan and KAVURI, Surya N: “A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies”. In: *Computers & chemical engineering* 27.3 (2003), pp. 313–326.
- [Ven03b] VENKATASUBRAMANIAN, Venkat; RENGASWAMY, Raghunathan; KAVURI, Surya N and YIN, Kewen: “A review of process fault detection and diagnosis: Part III: Process history based methods”. In: *Computers & chemical engineering* 27.3 (2003), pp. 327–346.
- [Ven03c] VENKATASUBRAMANIAN, Venkat; RENGASWAMY, Raghunathan; YIN, Kewen and KAVURI, Surya N: “A review of process fault detection and diagnosis: Part I: Quantitative model-based methods”. In: *Computers & chemical engineering* 27.3 (2003), pp. 293–311.
- [Ver22] VERMA, Thomas S and PEARL, Judea: “Equivalence and synthesis of causal models”. In: *Probabilistic and causal inference: The works of Judea Pearl*. 2022, pp. 221–236.
- [Wil20] WILLS, Peter and MEYER, François G: “Metrics for graph comparison: a practitioner’s guide”. In: *Plos one* 15.2 (2020), e0228728.
- [Wil76] WILLSKY, Alan S: “A survey of design methods for failure detection in dynamic systems”. In: *Automatica* 12.6 (1976), pp. 601–611.
- [Win77] WINNIE, John A: “The causal theory of space-time”. In: (1977).
- [Woo05] WOODWARD, James: *Making things happen: A theory of causal explanation*. Oxford university press, 2005.
- [Wri21] WRIGHT, Sewall: “Correlation and causation”. In: *Journal of Agricultural Research* (1921).
- [Yan22] YANG, Wenzhuo; ZHANG, Kun and HOI, Steven CH: “Causality-based multivariate time series anomaly detection”. In: *arXiv preprint arXiv:2206.15033* (2022).
- [Ye24] YE, Wenqian; ZHENG, Guangtao; CAO, Xu; MA, Yunsheng; HU, Xia and ZHANG, Aidong: “Spurious Correlations in Machine Learning: A Survey”. In: *arXiv preprint arXiv:2402.12715* (2024).

- [Yeh16] YEH, Chin-Chia Michael; ZHU, Yan; ULANOVA, Liudmila; BEGUM, Nurjahan; DING, Yifei; DAU, Hoang Anh; SILVA, Diego Furtado; MUEEN, Abdullah and KEOGH, Eamonn: “Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets”. In: *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE. 2016, pp. 1317–1322.
- [Zan22] ZANGA, Alessio; OZKIRIMLI, Elif and STELLA, Fabio: “A survey on causal discovery: theory and practice”. In: *International Journal of Approximate Reasoning* 151 (2022), pp. 101–129.
- [Zha08a] ZHANG, Jiji: “Causal reasoning with ancestral graphs”. In: *Journal of Machine Learning Research* 9 (2008), pp. 1437–1474.
- [Zha08b] ZHANG, Jiji and SPIRTES, Peter: “Detection of unfaithfulness and robust causal inference”. In: *Minds and Machines* 18.2 (2008), pp. 239–271. DOI: [10.1007/s11023-008-9096-4](https://doi.org/10.1007/s11023-008-9096-4).
- [Zha12a] ZHANG, Jiji and SPIRTES, Peter L: “Strong faithfulness and uniform consistency in causal inference”. In: *arXiv preprint arXiv:1212.2506* (2012).
- [Zha12b] ZHANG, Kun; PETERS, Jonas; JANZING, Dominik and SCHÖLKOPF, Bernhard: “Kernel-based conditional independence test and application in causal discovery”. In: *arXiv preprint arXiv:1202.3775* (2012).

Own publications

This section contains a complete list of own publications.

The publications [5], [7] address the topic of signal injections, the publications [9], [6], [10], [4] cover the topic of causal structure learning. While the publications [2], [11], [8], [12] deal with the topic of root cause identification. The publications [3], [1] can only be considered relevant to the present work in a broader sense.

- [1] GONSIOR, Julius; REHAK, Josephine; THIELE, Maik; KOCI, Elvis; GÜNTHER, Michael and LEHNER, Wolfgang: “Active Learning for Spreadsheet Cell Classification.” In: *EDBT/ICDT Workshops*. 2020.
- [2] KIMMIG, Andreas et al.: “Wertstromkinematik–Produktionssysteme neu gedacht”. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 116.12 (2021), pp. 935–939.
- [3] KOCI, Elvis; THIELE, Maik; REHAK, Josephine; ROMERO, Oscar and LEHNER, Wolfgang: “DECO: A dataset of annotated spreadsheets for layout and table recognition”. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE. 2019, pp. 1280–1285.
- [4] PFROMMER, Julius; REHAK, Josephine; ARABIZADEH, Negar and BEYERER, Jürgen: “Lattice-based Equivalence Classes for Causal Reasoning with Path-Constraints”. In: (2024).
- [5] REHAK, Josephine: “A Proposal on Discovering Causal Structures in Technical Systems by Means of Interventions”. In: *Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. 2021, p. 91.
- [6] REHAK, Josephine: “A Review on Approaches for Causal Structure Identification”. In: *Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. 2022, p. 105.
- [7] REHAK, Josephine: “Constraint-based Causal Discovery by using Path Constraints gained from Signal Injection and Recovery”. In: *Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. 2023, p. 47.
- [8] REHAK, Josephine: “Root Cause Analysis via Anomaly Detection and Causal Graphs”. In: *Dagstuhl Reports* 14 (2024).
- [9] REHAK, Josephine; FALKENSTEIN, Alexander and BEYERER, Jürgen: “Causal Structure Learning using PCMCi+ and Path Constraints from Wavelet-based Soft Interventions”. In: *Machine Learning for Cyber Physical Systems: Selected papers from the International Conference ML4CPS 2023*. Springer Nature, 2023.

- [10] REHAK, Josephine; FALKENSTEIN, Alexander and BEYERER, Jürgen: “Metrics for the Evaluation of Learned Causal Graphs Based on Ground Truth”. In: *Machine Learning for Cyber Physical Systems: Selected papers from the International Conference ML4CPS 2024*. 2024.
- [11] REHAK, Josephine; SOMMER, Anouk; BECKER, Maximilian; PFROMMER, Julius and BEYERER, Jürgen: “Counterfactual Root Cause Analysis via Anomaly Detection and Causal Graphs”. In: *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*. IEEE. 2023, pp. 1–7.
- [12] REHAK, Josephine; YOUSSEF, Shahenda and BEYERER, Jürgen: “Root Cause Analysis Using Anomaly Detection and Temporal Informed Causal Graphs”. In: *Machine Learning for Cyber Physical Systems: Selected papers from the International Conference ML4CPS 2024*. 2024.

Patents

This section contains patents that were created during my doctoral studies.

- [1] STÖRNER, Patrick; WOLL, Jürgen; PFROMMER, Julius; HASTEROK, Constanze and REHAK, Josephine: “Assistenzsystem zur automatischen Überwachung aller Prozesswerte und deren Korrelation”. DE102021206044A1. Mar. 1, 2020.

Supervised student theses

This section contains student work supervised during the doctoral program.

- [1] ALVARES, Bruno: “Accelerating PCMCI for Large Timeseries”. Master’s thesis. Karlsruhe: Karlsruhe Institute of Technology, Nov. 2022.
- [2] FALKENSTEIN, Alexander: “Constraint-based Causal Discovery on Simulations of Technical Systems”. Bachelor’s thesis. Karlsruhe: Karlsruhe Institute of Technology, Feb. 2022.
- [3] NEUBAUER, Tassilo: “Machine Learning-based Root Cause Analysis for Intelligent Production”. Bachelor’s thesis. Karlsruhe: Karlsruhe Institute of Technology, Oct. 2023.

List of Figures

2.1	An example DAG and its adjacency matrix representation.	7
2.2	An example CPDAG and its adjacency matrix representation.	7
2.3	An example MAG and its adjacency matrix representation.	8
2.4	An example PAG and its adjacency matrix representation.	8
2.5	Depicted is a simple example of a time series graph. It contains discrete nodes for each variable and for each sampled timestep $t_0, ..., t_4$	9
2.6	The corresponding summary graph for Figure 2.5.	9
2.7	Minimal example of a functional causal model	10
2.8	Simple building block of a causal graph with unoriented edges	13
2.9	Conditional independence tests applied to given data can identify immoralities in the graph structure.	13
2.10	Shown is the effect of a hard intervention interventions on a given causal structure. A hard interventions eliminates all other influences on the variable of intervention.	14
2.11	Shown is the effect of a soft interventions on a given causal structure. A soft intervention does not interfere with the existing causal structure.	14
2.12	The DAGs G_1 and G_2 are members of the equivalence class depicted by the CPDAG G_3 . The edge between a and b in G_3 is known to be present, but can be oriented in either direction.	16
2.13	The basic approach to evaluate learning method using ground truth	17
3.1	An example of how a Daubechie wavelet (a) and a sine-based wavelet with $f_{\text{sine}} = 100\text{Hz}$ (b) and $a = 1$ are injected and how they can travel to an observed variable (c,d) and how they may be rediscovered (e, f)	33
3.2	Depicted is the resized Daubechie wavelet ($a = 1$) that we chose for amplitude based injection and recovery	34
3.3	Depicted are the two signals we chose for frequency based injection and recovery	34
4.1	Truth tables of the ternary Kleene logic	39
4.2	A Binary decision diagram representing the Boolean function $f(x_1, x_2) = x_1 \vee x_2$. The root node is defined to be x_1	40
4.3	As example, we shown five ternary matrices. Each matrix $\mathbf{K}_3, \mathbf{L}_3, \mathbf{M}_3, \mathbf{N}_3$ is a refinement of \mathbf{O}_3	41
4.4	Alternative representations of the same equivalence class with four variables.	46
4.5	Simplified causal graph and altitudes of the Castleguard system.	47
4.6	Edge probability matrix of the Castleguard equivalence class	47

5.1	Overview of the nCEDs behavior for varying k on stepwise graph transformations of a five variable DAG if the transformation only affects one edge end of an undirected edge.	56
5.2	Overview of the nCEDs behavior for varying k on stepwise graph transformations of a five variable DAG if the transformation only affects both edge ends of an undirected edge. Since we want the condition $nCED(G_e, G^*) < nCED(G_{u,2}, G^*)$ to hold true, we can deduce from (c) that we need to choose $0 < k < 0.5$	57
5.3	The ground truth DAG and the created MAG for the Asia dataset. By eliminating the variable of smoking, we create a hidden confounder between lung infection and the bronchitis variable. The resulting graph is a ground truth MAG. The relevant structure differences are highlighted in green.	59
5.4	Example graphs as they were created by the different learning algorithms when applied on the adapted Asia dataset.	60
5.5	Overview of the behavior of the scoring methods given the sequential use of random structure changes on a five variable DAG. The scale of the SHD is shown on the plot's right side. We observe some scoring methods to have a gradient of zero and thereby to not be responsive to the induced changes. Of the responsive scoring methods, we observe most perform consistently indicated by a linear increase or decrease.	63
6.1	A general overview of the steps required for this RCA algorithm. We point out the steps we contribute to in this chapter via solid frames.	68
6.2	Given the shown example sets $\mathcal{V}_G(r_1)$ and \mathcal{V}_A for root cause r_1 , the similarity between the variable sets are scored as $JRCS(r_1) = 0.5$	70
6.3	Shown are the basic process steps of the gripping simulation. There are proximity sensors located in the shelf and on the belt. The images depict collaborating robots as they were another research topic of the funding project [Müh21, Kim21].	74
6.4	The summary causal graph of the robotic setup. The horizontal orientation of the variables indicates their affiliation with the variable sets shown above. Observed values appearing several times are identical. Required sets of process steps of a causal relation are annotated to the edges of the graph.	75

List of Tables

2.1	In tabular data, the number of possible graphs grows exponentially with the number of variables N	10
3.1	Given a bivariate FCM, if we inject a signal w in a in a bivariate scenario, our signal w is part of the computation of b only if $a \rightarrow b$. If $a \perp\!\!\!\perp b$ or $a \leftarrow b$ holds, then w is not part of the computation of b	22
3.2	Given a trivariate FCM, if we inject a signal w in a via $\text{do}(a, w)$, we may observe traces of the signal w to be observed in variables depending on a (Case I) but also in variables indirectly depending on a (Case II and III).	24
3.3	If we inject a signal w in a in a bivariate scenario and we recover the remainder of w in b , then we know b is successor of a . Since we consider the bivariate case, we infer $a \rightarrow b$	25
3.4	In a multivariate scenario, we may identify present paths from the recovery of the w and we may identify the corresponding absent edges if we assume the graph to be a acyclic.	26
3.5	If we assume the summary graphs to be acyclic and assume the variables to not be autoregressive, we may learn the constraints from the example cases as shown. Here, the number below the summary graph variables indicate the order of signal recovery.	28
3.6	If we allow cycles in the summary graph and allow variables to be autoregressive, we may learn the constraints from the example cases as shown. Here again, the number below the graph variables, indicate the order of signal recovery.	30
3.7	Results of the signal injection and recovery for different amplitudes. Each entry is the average and standard deviation of ten PRC AUC measurements on ten generated datasets.	35
4.1	Number of binary and ternary adjacency matrices required to represent the constrained equivalence classes for growing number of variables N . This table was produced using the Cudd library [Som98].	42
4.2	Calculation time and number of required BDD nodes for the shown constraints. Temporal measurements were taken by measuring ten times and by calculating the average. The BDD results were produced by cudd library [Som98].	44
4.3	Listed are the sites of tracer injection and the sampling sites where the tracers were discovered in the simplified Karst Rock example.	47

4.4	Number of required conditional independence tests for PC1 and PCPC1 on benchmark examples for causal identification. We used the Fisher z-independence test with a threshold of 0.05. The shown values are averages and standard deviation for 20 runs each.	49
4.5	Execution times of PCPC1 on two example datasets with limited affected path lengths k . Consider, that for $K = 1$ PCPC1 is identical to the PC1 algorithm. Shown are the averages and standard deviations over ten samples each.	50
5.1	Results of the nCED with a MAG as ground truth for graphs learned on the adapted Asia dataset. The lower the score, the better performed the structure learning algorithm.	59
5.2	Two examples of how the examined scoring methods evaluate causal graphs learned by the HC), PC and FCI structure learning algorithm. The higher the scores of the ROC AUC, PRC AUC, F_1 , TPR and FPR, the better the algorithm performed. For the SHD and nCED variants, lower scores are better. (* The scoring method only considers present and absent directed edges, unaware of other edge types.)	62
5.3	Bounds and normalizability of the inspected scoring methods	64
6.2	Overview of the potential anomalous observations affected by the root causes in the shown process steps. Each row corresponds to $\mathcal{T}_G(r)$ for its respective root cause.	76
6.1	Overview of potential root causes and their categories. Also, it is indicated by whom the prior knowledge about the root cause may be provided as stated by the interviewed domain experts	77
6.3	Overview of the discovered anomalous observations and their corresponding process steps. Each row corresponds to \mathcal{T}_A for its respective scenario.	78
6.4	Jaccard Root Cause Scores for both simulated anomalous scenarios and each root cause for developed approaches. A value of one indicates an ideal match between the root causes and the detected anomalies. We observe better distinction between the root causes, if temporal information is considered.	78

Acronyms

AI	Artificial Intelligence
ATE	Average Treatment Effect
BDD	Binary Decision Diagram
CD	Causal Discovery
CED	Causal Edit Distance
CIT	Conditional Independence Test
CPDAG	Complete Partially Directed Acyclic Graph
CSL	Causal Structure Learning
DAG	Directed Acyclic Graph
FCI	Fast Causal Inference
FCM	Functional Causal Model
FFT	Fast Fourier Transform
FPR	False Positive Rate
HCS	Hill Climb Search algorithm
JRCS	Jaccard Root Cause Score
MAG	Maximal Ancestral Graph
MEC	Markov Equivalence Class
nCED	normalized Causal Edit Distance
PAG	Partial Ancestral Graph
PC	Peter-Clark algorithm
PCPC1	Path Constrained Peter-Clark algorithm
PRC AUC	Precision Recall Curve Area Under Curve
RCA	Root Cause Analysis
ROC AUC	Receiver Operating Characteristic Area Under Curve

SCM	Structural Causal Model
SHD	Structural Hamming Distance
TPR	True Positive Rate

