Hasso-Plattner-Institut Für Digital Engineering
Enterprise Platform And Integration Concepts

Causal Discovery in Practice: Non-Parametric Conditional Independence Testing and Tooling for Causal Discovery

Dissertation
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)
in der Wissenschaftsdisziplin *Praktische Informatik*

eingereicht an der
Digital Engineering Fakultät
der Universität Potsdam

von

Johannes Eric Hügle

Betreuer:
Prof. Dr. h.c. mult. Hasso Plattner
Prof. Dr. Tilmann Rabl

Gutachter:
Prof. Dr. Jakob Runge
Prof. Dr. Kristian Kersting

Potsdam, 9. Oktober 2023

Published online on the
Publication Server of the University of Potsdam:
https://doi.org/10.25932/publishup-63582
https://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-635820

Abstract

Knowledge about causal structures is crucial for decision support in various domains. For example, in discrete manufacturing, identifying the root causes of failures and quality deviations that interrupt the highly automated production process requires causal structural knowledge. However, in practice, root cause analysis is usually built upon individual expert knowledge about associative relationships. But, “correlation does not imply causation”, and misinterpreting associations often leads to incorrect conclusions. Recent developments in methods for causal discovery from observational data have opened the opportunity for a data-driven examination. Despite its potential for data-driven decision support, omnipresent challenges impede causal discovery in real-world scenarios. In this thesis, we make a threefold contribution to improving causal discovery in practice.

(1) The growing interest in causal discovery has led to a broad spectrum of methods with specific assumptions on the data and various implementations. Hence, application in practice requires careful consideration of existing methods, which becomes laborious when dealing with various parameters, assumptions, and implementations in different programming languages. Additionally, evaluation is challenging due to the lack of ground truth in practice and limited benchmark data that reflect real-world data characteristics.

To address these issues, we present a platform-independent modular pipeline for causal discovery and a ground truth framework for synthetic data generation that provides comprehensive evaluation opportunities, e.g., to examine the accuracy of causal discovery methods in case of inappropriate assumptions.

(2) Applying constraint-based methods for causal discovery requires selecting a conditional independence (CI) test, which is particularly challenging in mixed discrete-continuous data omnipresent in many real-world scenarios. In this context, inappropriate assumptions on the data or the commonly applied discretization of continuous variables reduce the accuracy of CI decisions, leading to incorrect causal structures.

Therefore, we contribute a non-parametric CI test leveraging k-nearest neighbors methods and prove its statistical validity and power in mixed discrete-continuous data, as well as the asymptotic consistency when used in constraint-based causal discovery. An extensive evaluation of synthetic and real-world data shows that the proposed CI test outperforms state-of-the-art approaches in the accuracy of CI testing and causal discovery, particularly in settings with low sample sizes.

(3) To show the applicability and opportunities of causal discovery in practice, we examine our contributions in real-world discrete manufacturing use cases. For example, we showcase how causal structural knowledge helps to understand unforeseen production downtimes or adds decision support in case of failures and quality deviations in automotive body shop assembly lines.

Zusammenfassung

Kenntnisse über die Strukturen zugrundeliegender kausaler Mechanismen sind eine Voraussetzung für die Entscheidungsunterstützung in verschiedenen Bereichen. In der Fertigungsindustrie beispielsweise erfordert die Fehler-Ursachen-Analyse von Störungen und Qualitätsabweichungen, die den hochautomatisierten Produktionsprozess unterbrechen, kausales Strukturwissen. In Praxis stützt sich die Fehler-Ursachen-Analyse in der Regel jedoch auf individuellem Expertenwissen über assoziative Zusammenhänge. Aber "Korrelation impliziert nicht Kausalität", und die Fehlinterpretation assoziativer Zusammenhänge führt häufig zu falschen Schlussfolgerungen. Neueste Entwicklungen von Methoden des kausalen Strukturlernens haben die Möglichkeit einer datenbasierten Betrachtung eröffnet. Trotz seines Potenzials zur datenbasierten Entscheidungsunterstützung wird das kausale Strukturlernen in der Praxis jedoch durch allgegenwärtige Herausforderungen erschwert. In dieser Dissertation leisten wir einen dreifachen Beitrag zur Verbesserung des kausalen Strukturlernens in der Praxis.

(1) Das wachsende Interesse an kausalem Strukturlernen hat zu einer Vielzahl von Methoden mit spezifischen statistischen Annahmen über die Daten und verschiedenen Implementierungen geführt. Daher erfordert die Anwendung in der Praxis eine sorgfältige Prüfung der vorhandenen Methoden, was eine Herausforderung darstellt, wenn verschiedene Parameter, Annahmen und Implementierungen in unterschiedlichen Programmiersprachen betrachtet werden. Hierbei wird die Evaluierung von Methoden des kausalen Strukturlernens zusätzlich durch das Fehlen von "Ground Truth" in der Praxis und begrenzten Benchmark-Daten, welche die Eigenschaften realer Datencharakteristiken widerspiegeln, erschwert.

Um diese Probleme zu adressieren, stellen wir eine plattformunabhängige modulare Pipeline für kausales Strukturlernen und ein Tool zur Generierung synthetischer Daten vor, die umfassende Evaluierungsmöglichkeiten bieten, z.B. um Ungenauigkeiten von Methoden des Lernens kausaler Strukturen bei falschen Annahmen an die Daten aufzuzeigen.

(2) Die Anwendung von constraint-basierten Methoden des kausalen Strukturlernens erfordert die Wahl eines bedingten Unabhängigkeitstests (CI-Test), was insbesondere bei gemischten diskreten und kontinuierlichen Daten, die in vielen realen Szenarien allgegenwärtig sind, die Anwendung erschwert. Beispielsweise führen falsche Annahmen der CI-Tests oder die Diskretisierung kontinuierlicher Variablen zu einer Verschlechterung der Korrektheit der Testentscheidungen, was in fehlerhaften kausalen Strukturen resultiert.

Um diese Probleme zu adressieren, stellen wir einen nicht-parametrischen CI-Test vor, der auf Nächste-Nachbar-Methoden basiert, und beweisen dessen statistische Validität und Trennschärfe bei gemischten diskreten und kontinuierlichen Daten, sowie dessen asymptotische Konsistenz in constraint-basiertem kausalem Strukturlernen. Eine umfangreiche Evaluation auf synthetischen und realen Daten zeigt, dass der vorgeschlagene

CI-Test bestehende Verfahren hinsichtlich der Korrektheit der Testentscheidung und gelernter kausaler Strukturen übertrifft, insbesondere bei geringen Stichprobengrößen.

(3) Um die Anwendbarkeit und Möglichkeiten kausalen Strukturlernens in der Praxis aufzuzeigen, untersuchen wir unsere Beiträge in realen Anwendungsfällen aus der Fertigungsindustrie. Wir zeigen an mehreren Beispielen aus der automobilen Karosseriefertigungen wie kausales Strukturwissen helfen kann, unvorhergesehene Produktionsausfälle zu verstehen oder eine Entscheidungsunterstützung bei Störungen und Qualitätsabweichungen zu geben.

To my family.

Contents

Part I Tooling for Causal Discovery

Part II Causal Discovery from Mixed Discrete-Continuous Data

Part III Causal Discovery in Practice and Conclusion

1

Introduction

In this opening chapter, we motivate this thesis with the example of occurring failures and quality deviations that interrupt an automotive manufacturing process (Section 1.1). In this context, causal discovery, for which we provide a gentle introduction (Section 1.2), has attracted increasing attention as a basis for data-driven decision support. Although methods for causal discovery enable the root cause analysis of failures and quality deviations, multiple challenges impede its application in practice (Section 1.3). In this thesis, we contribute in a threefold manner: (1) we provide the tooling necessary to support the evaluation and the applicability of methods for causal discovery, (2) we introduce a non-parametric conditional independence (CI) test that improves causal discovery from mixed discrete-continuous data, and (3), we demonstrate the applicability and show opportunities in real-world discrete manufacturing scenarios (Section 1.4). Further, we outline the structure of this thesis (Section 1.5).

1.1 Motivation from Automotive Manufacturing

Automotive manufacturing enterprises must cope with growing demands for increased product quality, greater product variability, reduced cost, and global competition [97]. To meet these demands, modern car body shop assembly lines are highly optimized and operative with a minimum human intervention [54].

As depicted in Fig. 1.1 (on page 2), an assembly line consists of multiple sections, e.g., underbody, body assembly, attachments, and finish, where each section is separated into high-automated production cells. In this context, hundreds of individual steel and aluminum parts are assembled step by step to form the vehicles' metal coats. For example, in the attachments section, doors, bonnets, and boot lids are added to the vehicles' bodies by robots that weld, rivet, or bend in the high-automated production cells. The occurrence of failures and deviations of quality measurements are a major cause of unscheduled stoppage of the car body assembly line and are costly not only in terms of time lost but also in terms of capital destroyed [19]. Despite the high degree of automation, human workers are essential for quality control, systems operation, and root cause analysis of unscheduled stoppages. Usually, the analysis of failures and quality deviations is built upon non-persistent, individual on-site expert knowledge, and hence, troubleshooting relies on the individual knowledge of the staff on shift [70]. Therefore, the technical staff relies on their experience with coincidences of failures and quality deviations, which may be error-prone.

To provide a simplified example that serves as a running example, we will consider a welding robot within a production cell of a car body shop assembly line. Preventing bad weld seams is crucial for the quality of automotive manufacturing, such that the technical staff constantly observes the robots' welding process. As depicted in Fig. 1.2, there is an

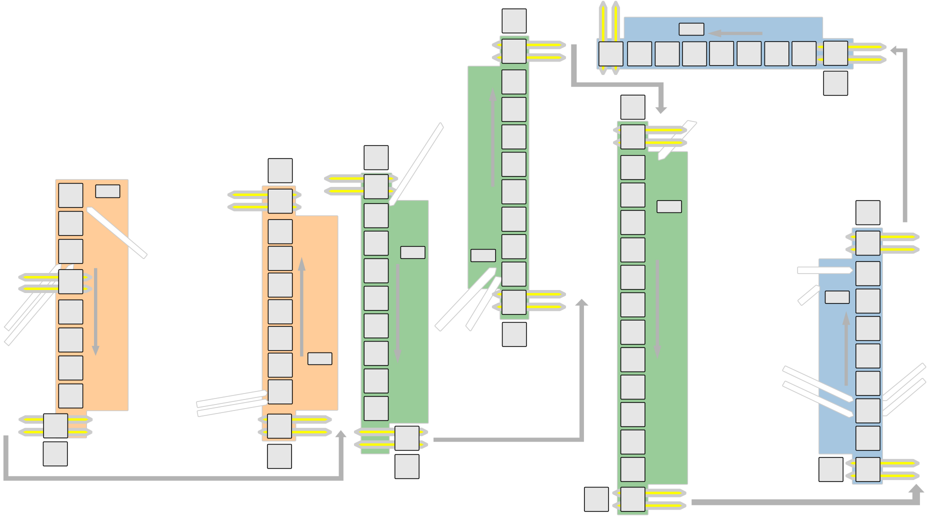
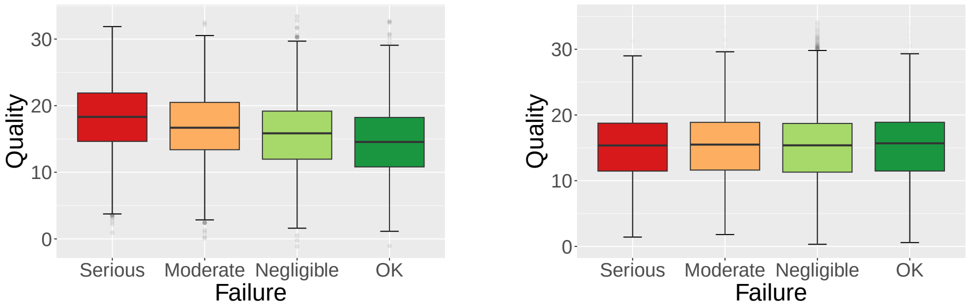


Fig. 1.1: A schematic car body shop assembly line consisting of multiple sections, i.e., underbody (orange), body assembly (green), and attachments and finish (blue), where each section is separated into high-automated production cells (grey squares) in which robots weld, rivet, or bend.

observable coincidence of failures ("Serious", "Moderate", "Negligible", "OK") sent by the welding robot and quality deviations of the weld seam (smaller better), denoted by $Failure$ and $Quality$, respectively. In particular, more serious errors relate to higher deviations in the quality of the weld seam, i.e., $Quality$ and $Failure$ are dependent, denoted by $Quality \not\perp\!\!\!\perp Failure$ (see Fig. 1.2 (a), left). Hence, one may conclude that preventing failures in the welding process results in decreased quality deviations, i.e., assuming that $Failure$ causally influences $Quality$. But "*correlation does not imply causation*" and misinterpreting associative behaviors between variables yields incorrect causal conclusions. In particular, preventing failures in our welding example will not affect the quality of a



(a) $Quality \not\perp\!\!\!\perp Failure$ **(b)** $Quality \perp\!\!\!\perp Failure \mid Temperature$

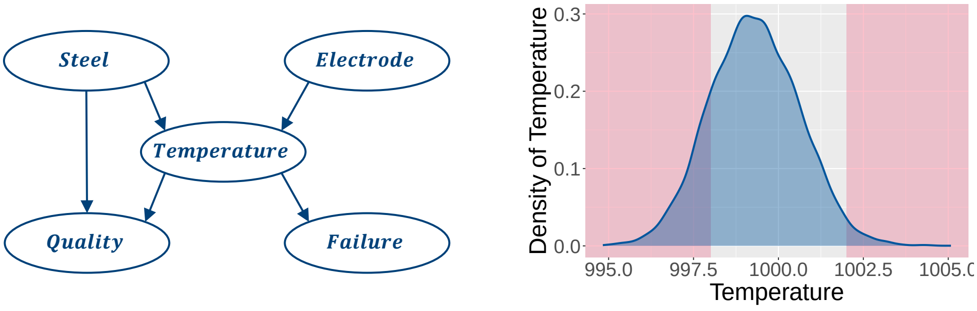
Fig. 1.2: (Confounder) Boxplots depicting the relationship between $Quality$ (smaller better) and $Failure$ of a welding process. The observational data shows a dependence between $Quality$ and $Failure$, i.e., higher quality deviations are related to more serious errors; see (a). However, the dependence vanishes conditioned on the confounding common cause $Temperature$, i.e., similar distributions of quality deviations for all errors given $Temperature$ around 1 000 degrees; see (b).

weld seam! This is due to the confounding $Temperature$ of the welding process that is the decisive factor causally influencing the quality and failures of the welding process. This observation follows Reichenbach's common cause principle [142], which states that whenever there is a relation between two variables, such as $Failure$ and $Quality$, then either, first, $Failure$ causally influences $Quality$, second, $Quality$ causally influences $Failure$, or third, there is a common cause influencing both. Thus, conditioned on the confounding common cause $Temperature$, the dependence of $Quality$ and $Failure$ vanishes given $Temperature$ around 1 000 degrees, i.e., $Quality$ and $Failure$ become independent conditioned on $Temperature$. Hence, we observe that $Quality \perp\!\!\!\perp Failure \,|\, Temperature$ (see Fig. 1.2 (b) on page 2, right).

Note that similar incorrect conclusions are common pitfalls in the application of machine learning or deep learning techniques, which build upon the associative nature of probability theory, e.g., see [130, 168]. Hence, the Turing award-winning Judea Pearl points out that "*Machines' lack of understanding of causal relations is perhaps the biggest roadblock to giving them human-level intelligence*" [131]. In this context, the emergence of methods for causal discovery, e.g., see [168, 130], created the basis of a data-driven assessment of the causal relationships from observational data of a manufacturing process [95, 108, 54, 70].

1.2 A Gentle Introduction to Causal Discovery

Generally, *causal discovery*, also referred to as *causal structure learning (CSL)*, aims to infer the underlying data-generating causal structures among a set of variables $\mathbf{V}$, e.g., of relevant features of the manufacturing process in an automotive assembly line, i.e., $\mathbf{V} = \{Steel, Electrode, Temperature, Quality, Failure\}$. In this context, the underlying causal structures are depicted in a *causal graph* $\mathcal{G}$ where each node represents a variable, e.g., $Temperature$ or $Failure$, and a directed edge between two nodes represents a direct causal relationship, e.g., $Temperature \rightarrow Quality$ denotes that $Temperature$ is the cause of $Quality$. In our exemplary automotive assembly line, Fig. 1.3 (a) depicts the underlying causal structures of the welding process as a causal graph $\mathcal{G}$ that serves as a running example in this gentle introduction. These structures are not, or only partially, known to the technical staff on a real automotive assembly line. Therefore, we aim to infer as much of these causal structures from observational data of the welding process as possible.



(a) Causal structures in $\mathcal{G}$

(b) Density of $Temperature$

Fig. 1.3: The welding process is determined by the underlying causal structures between $Steel$, $Electrode$, $Temperature$, $Quality$, and $Failure$; see (a). The continuous distributed welding temperature $Temperature$ is the crucial factor of the welding process with an optimal value range of around 1 000 degrees with critical temperature ranges highlighted in red; see (b).

The crucial factor of the welding process is the applied temperature at the welding electrodes, which follows a continuous distribution with an optimal value range of around 1 000 degrees as depicted in Fig. 1.3 (b) (on page 3). It is directly influenced by the grade of the welded steel, i.e., $Steel \rightarrow Temperature$, where $Steel$ is categorized according to the steel's chemical composition as “Excellent”, “Normal”, or “Poor”. Further, the technician controls the welding process by adjusting the electrodes' setting, i.e., $Electrode \rightarrow Temperature$, where $Electrode$ is a continuously distributed value centered around zero. As previously observed in Section 1.1, the temperature causally influences both the quality of the welding seam, i.e., $Temperature \rightarrow Quality$, and whether the welding process can be carried out or failed due to errors, i.e., $Temperature \rightarrow Failure$. Further, the quality of the welding seam depends on the quality of the steel, i.e., $Steel \rightarrow Quality$.

The causal structures depicted in Fig. 1.3 (a) (on page 3) are an accessible graphical representation of the more complicated underlying causal mechanisms. For example, in $Steel \rightarrow Temperature$, the discrete values of $Steel$ incorporate variations of carbon content, inclusions, or conductivity of the different steel grades, which influences the welding temperature following complicated thermodynamic processes. In a real automotive assembly line, there also exist other factors that influence the welding temperature, e.g., small fluctuations of the factory's temperature, the component's geometry, and many more. In our causal model, we abstract those away as noise and model the welding temperature as

$$Temperature = f(Steel, Electrode, N_T),$$

where f is a function of $Steel$, $Electrode$, and an independent Gaussian distributed noise variable N_T. Similarly, all variables in $\mathbf{V}$ can be modeled as functions of their causes, which yield a *structural causal model (SCM)* defining the causal mechanisms of the welding process by

$$
\begin{aligned}
&Steel = N_S, && N_S \sim \mathcal{B}(2, 0.5) && (1.1)\\
&Electrode = N_E, && N_E \sim \mathcal{N}(0, 1) && (1.2)\\
&Temperature = 0.3 * Electrode^2 - Steel + N_T, && N_T \sim \mathcal{N}(1000, 1) && (1.3)\\
&Quality = Temperature + 7 * Steel + N_Q, && N_Q \sim \mathcal{N}(-990, 3) && (1.4)\\
&Failure = f_F(Temperature) + N_F, && N_F \sim \mathcal{B}(3, 0.2), && (1.5)
\end{aligned}
$$

where f_F is a probabilistic mapping from $\mathbb{R}$ to $\mathbb{Z}/4\mathbb{Z}$. In a real automotive assembly line, SCMs are generally unknown to the technical staff, but they abstract the underlying thermodynamic processes. As defined above, the variables that determine our welding process are either discrete distributed, such as $Steel$ and $Failure$ as defined in Eq. (1.1) and (1.5), continuous distributed, such as $Electrode$ as defined in Eq. (1.2), or follow a discrete-continuous mixture distribution, such as $Temperature$ and $Quality$ as defined in Eq. (1.3) and (1.4). In this context, the discrete realizations of $Steel$ and $Failure$ relate to the ordinal values “Excellent”, “Normal”, or “Poor” and “Serious”, “Moderate”, “Negligible”, “OK”, respectively. Hence, from Eq. (1.1), we see that poorer steel grades yield lower welding temperatures, as graphically depicted in the edge $Steel \rightarrow Temperature$ in $\mathcal{G}$; see Fig. 1.3 (a) (on page 3).

Altogether, this SCM allows us to model the nodes of the causal graph $\mathcal{G}$ (see Fig. 1.3 (a) on page 3) as random variables $\mathbf{V}$ and defines a joint distribution $P_{\mathbf{V}}$ among them. The *observational data*, i.e., realizations of $\mathbf{V}$ are independent and identical (i.i.d.) distributed samples drawn from $P_{\mathbf{V}}$. In our running example, the observational data of the welding process is shown in Table 1.1 (on page 5) being synthetically generated according to the SCM. Hence, the goal of causal discovery is to learn the underlying causal structures of $\mathcal{G}$ from the observational data that can be, in our case synthetically, gathered during the welding process. However, “*No causation without manipulation.*” [62],

Table 1.1: Observational data of our welding process example, which is sampled according to the SCM defined in Eq. (1.1) - (1.5). It comprises i.i.d. samples of all relevant variables $\mathbf{V} = \{Steel, Electrode, Temperature, Quality, Failure\}$ drawn according to the joint distribution $P_{\mathbf{V}}$.

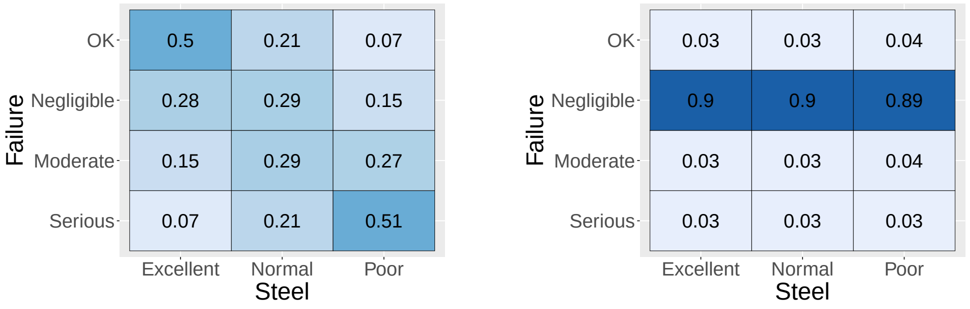
Electrode	***Steel***	***Temperature***	***Quality***	***Failure***
-0.5604756	Excellent	1001.5476	0.02271	OK
-0.2301775	Normal	997.5680	6.03202	OK
1.5587083	Poor	999.1471	25.76893	Moderate
0.0705084	Poor	997.0821	20.78925	Serious
0.1292877	Normal	1002.7410	2.14548	Negligible
0.4609162	Poor	995.9723	9.33221	Serious
-1.2650612	Poor	998.3094	22.22655	Serious
-0.6868529	Normal	998.8192	4.01134	Moderate
1.7150650	Poor	998.0470	18.89567	Serious
-0.4456620	Normal	999.2840	13.27954	Moderate
...	...	...	...	...

such that, unlike data collected through controlled experiments, observational data is insufficient to infer causal structures [130]. Yet, if we are willing to make some assumptions, ideally as mild as possible, such that they are likely to hold in practice, causal discovery becomes feasible [168].

A common assumption is called *causal sufficiency*, which assumes that we observe all relevant variables, i.e., there are no unmeasured confounders [168]. Besides, we will assume that the causal structures in the causal graph are *acyclic*, i.e., there are no feedback loops. Under the assumption that causal sufficiency holds and there are no cycles, the true causal graph can be described by a *directed acyclic graph (DAG)* $\mathcal{G}$, similar to the one shown in Fig. 1.3 (a) on page 3. Further, to infer the DAG $\mathcal{G}$ over nodes $\mathbf{V}$ from i.i.d. samples drawn according to $P_{\mathbf{V}}$, we need to make assumptions that hold for the graph and the joint distribution. Most common are the *causal Markov condition (CMC)* and the *faithfulness* assumption, which state that the graphical separation in the causal structures of $\mathcal{G}$ yields conditional independence (CI) characteristics of the joint distribution $P_{\mathbf{V}}$ and vice versa [168].

Roughly, the CMC states that conditioned on its causes in the causal graph $\mathcal{G}$, every variable is independent of every other variable according to $P_{\mathbf{V}}$, except its effects. For example, consider our introductory example on $Quality$ and $Failure$ that have a common cause $Temperature$ in $\mathcal{G}$, i.e., $Quality \leftarrow Temperature \rightarrow Failure$. Hence, we can correctly conclude from the CMC that $Quality$ is independent of $Failure$ if we condition on $Temperature$, i.e., $Quality \perp\!\!\!\perp Failure \,|\, Temperature$; see Fig. 1.2 (b) on page 2. In contrast, $Quality$ and $Failure$ are dependent, if we do not condition on $Temperature$, i.e., $Quality \not\perp\!\!\!\perp Failure$; see Fig. 1.2 (a). Similarly, $Steel$ and $Failure$ are connected through a chain over $Temperature$ in $\mathcal{G}$, i.e., $Steel \rightarrow Temperature \rightarrow Failure$. Hence, the CMC implies that the dependence of $Steel$ and $Failure$ vanishes if we condition on the $Failures$ cause $Temperature$; see Fig. 1.4 (b) on page 6. However, since $Steel$ and $Failure$ are connected through a directed path, it may still hold that $Steel$ and $Failure$ are dependent if we do not condition on $Temperature$; see Fig. 1.4 (a).

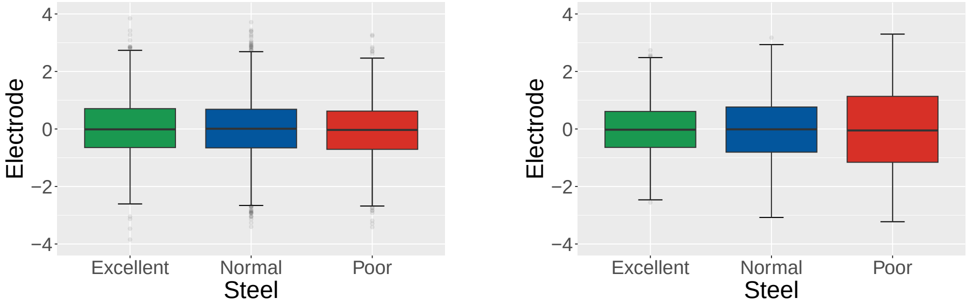
The faithfulness assumption simplified means that whatever (in)dependency occurs in $P_{\mathbf{V}}$, it arises not from incredible coincidence but rather from the structure of the DAG $\mathcal{G}$. For example, $Steel$ and $Electrode$ are, obviously, independent in $P_{\mathbf{V}}$ as both variables are generated through the SCM from two independent noise variables N_S and



(a) $Steel \not\perp\!\!\!\perp Failure$ **(b)** $Steel \perp\!\!\!\perp Failure \,|\, Temperature$

Fig. 1.4: (Chain) Heatmaps depicting the relationships between $Steel$ and $Failure$ within our welding process, with color grading according to the samples' proportions (rounded to two decimal digits). The observational data shows a dependence between $Steel$ and $Electrode$, i.e., different frequencies of errors for different steel grades, such as higher frequencies of serious errors for poor steel grades; see (a). However, the dependence vanishes conditioned on $Temperature$, which is the cause of $Failure$ in the chain $Steel \rightarrow Temperature \rightarrow Failure$, i.e., equal frequencies of errors for all steel grades given a $Temperature$ around 1 000 degrees; see (b).

N_E, see Eq. (1.1) and (1.2) on page 4, respectively. Hence, as depicted in Fig. 1.5 (a), the observational data of our welding process shows independence between $Steel$ and $Electrode$, too. Therefore, we correctly conclude from the faithfulness assumption that both nodes are not adjacent in the true causal graph $\mathcal{G}$; see Fig. 1.3 (a) on page 3. From the graph $\mathcal{G}$, on the other hand, we see that all directed paths from $Steel$ to $Electrode$ are blocked by the node $Temperature$, which is a collider, i.e., $Steel \rightarrow Temperature \leftarrow Electrode$. Accordingly, we expect that due to the CMC $Steel$ and $Electrode$ will become independent if we condition on $Temperature$ as depicted in Fig. 1.5 (b).



(a) $Steel \perp\!\!\!\perp Electrode$ **(b)** $Steel \not\perp\!\!\!\perp Electrode \,|\, Temperature$

Fig. 1.5: (Collider) Boxplots depicting the relationships between $Steel$ and $Electrode$ in our welding process. The observational data shows independence between $Steel$ and $Electrode$, i.e., similar distributions of electrode settings for all steel grades; see (a). However, the dependence vanishes if we condition on the colliding common effect $Temperature$ in $Steel \rightarrow Temperature \leftarrow Electrode$, i.e., high variance of electrode settings are related to poorer steel qualities given $Temperature$ around 1 000 degrees; see (b).

In summary, the causal structures between variables $\mathbf{V}$ are encoded in a causal graphical model (CGM) consisting of the corresponding DAG $\mathcal{G}$, and the joint distribution over $\mathbf{V}$, e.g., see [130, 168].

After briefly discussing the background on SCMs, the CMC, and the faithfulness assumption, we will sketch how to learn the causal structures from observational data under those assumptions. In particular, assuming that both assumptions hold, we can 1) distinguish between (in)dependencies for each node, that is, infer the true undirected graph, and 2) orient some of the edges. Note that this introduction is tailored towards *constraint-based causal discovery*, which relies on applying conditional independence (CI) tests. However, the main ideas also translate to other classes of algorithms, such as score-based methods. The most popular constraint-based algorithm for causal discovery is the well-known Peter and Clark (PC) algorithm [169], mainly considered in this thesis. Under the assumption that the true underlying causal structures in $\mathcal{G}$ are acyclic and that causal sufficiency, faithfulness, and CMC hold, the PC algorithm achieves 1) and 2) as depicted in Fig. 1.6 according to the following main ideas. First, to obtain the undirected graph, called *skeleton*, the algorithm starts with a fully connected graph, i.e., each pair of nodes is connected through an edge (see Fig. 1.6 left).

First, the PC algorithm deletes edges based on the faithfulness assumption; see Fig. 1.6, 1). In particular, edges between two nodes are deleted if they can be rendered independent by testing for conditional independence given a set of other random variables. For example, in the welding process, we can delete the edge between $Steel$ and $Electrode$ since $Steel \perp\!\!\!\perp Electrode$; see Fig. 1.5 (a). Further, we can delete the edges $Electrode$ — $Quality$ and $Steel$ — $Failure$ as they are blocked by $Temperature$ in the corresponding chains $Electrode \rightarrow Temperature \rightarrow Quality$ and $Steel \rightarrow Temperature \rightarrow Failure$ which yields conditional independence in $P_{\mathbf{V}}$, respectively, e.g., see Fig. 1.4 (b). Finally, we can delete the edge between $Quality$ and $Failure$ since $Quality \perp\!\!\!\perp Failure \,|\, Temperature$ for the confounding $Temperature$; see Fig. 1.2 (b) on page 2. Generally, this first part of the PC algorithm, called skeleton discovery, iterates over the adjacency structures of all nodes until no further independence can be detected.

Second, to infer some of the edge directions, see Fig. 1.6, 2), it suffices to identify v-structures. A v-structure describes a triple of nodes in which two non-adjacent nodes jointly cause the third colliding node, e.g., $Steel \rightarrow Temperature \leftarrow Electrode$. In particular, if faithfulness holds, $Steel$ and $Electrode$ are dependent given $Temperature$, even if we additionally condition on any other node in the graph; see Fig. 1.5 (b). Vice versa, $Steel$ and $Electrode$ can be rendered independent if we do not condition on $Temperature$, see Fig. 1.5 (a), such that we can identify this v-structure, and, hence infer the corresponding edge directions. The PC algorithm repeats this procedure for each unshielded triple in the skeleton, which we determined in the previous step. As a

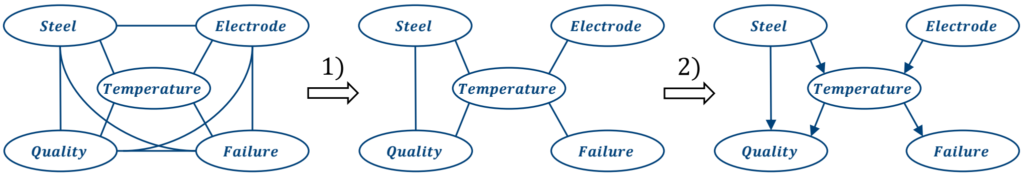


Fig. 1.6: (PC algorithm) First, see 1), the PC algorithm starts with a fully connected graph (left) and deletes edges through the application of CI tests, which yields the skeleton of $\mathcal{G}$ (center). Second, see 2), the identification of v-structures and the application of Meek's rules allow orienting edges to return the CPDAG of $\mathcal{G}$ (right).

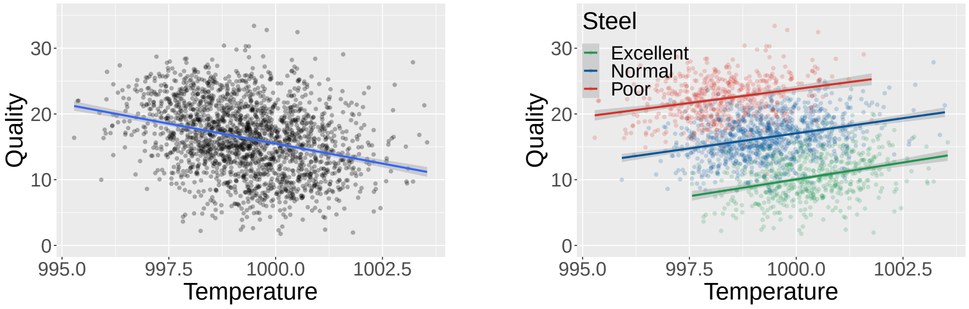
result, we identified the correct skeleton and all v-structures. Subsequently, it might be possible to further infer some edge directions due to the graph's acyclicality by applying Meek's rules [115]. Such a *complete partially directed acyclic graph (CPDAG)* represents the Markov equivalence class of the true DAG $\mathcal{G}$, provided that faithfulness and the CMC hold. Generally, the PC algorithm allows only to infer the edge directions up to the Markov equivalence class and needs to leave some of the edges undirected. In our exemplary welding process, the CPDAG coincides with the true DAG $\mathcal{G}$, i.e., there is only one graph $\mathcal{G}$ in the Markov equivalence class, such that the PC algorithm unambiguously returns the true $\mathcal{G}$ (Fig. 1.6 right). In particular, after detecting the v-structure $Steel \rightarrow Temperature \leftarrow Electrode$, we can further orient $Temperature \rightarrow Failure$ and $Temperature \rightarrow Quality$ as an inverse direction would generate the non-existing v-structures $Steel \rightarrow Temperature \leftarrow Failure$ and $Electrode \rightarrow Temperature \leftarrow Qualtiy$, respectively. Finally, we can orient $Steel \rightarrow Quality$ as an inverse direction would generate the cycle $Steel \rightarrow Temperature \rightarrow Quality \rightarrow Steel$, which contradicts the acyclicality of $\mathcal{G}$.

In summary, causal discovery allows learning the underlying causal structures from the observational data of the welding process. Let's recapitulate that the technician aims to improve the quality of the weld seams; see Section 1.1. In this context, the causal structures in $\mathcal{G}$ show that $Quality$ is directly causally influenced solely by $Steel$ and $Temperature$; see Fig. 1.6 (right). Hence, impeding serious errors will not affect $Quality$; see Fig. 1.2. As the technician cannot change the steel grade, $Temperature$ becomes the crucial factor in controlling the welding process to achieve quality improvements. In this context, the CGM is the key to formalizing causality as established in the groundbreaking work of Judea Pearl, for which he received the 2011 Turing Award. Traditionally, examining how the variable $Temperature$ causally influences $Quality$ is built upon randomized experiments or interventions into the system under observation, i.e., manually changing the $Temperature$ even if this may break the welding system. This ensures that observed changes in $Quality$ cannot be associated with changes of other variables but are solely implied through $Temperature \rightarrow Quality$. Such an intervention on $Temperature$ changes $Temperature$ to a fixed temperature $temperature$, which graphically matches the deletion of all incoming edges of $Temperaute$ in $\mathcal{G}$ transforming the observational to an experimental setup [130]. This concept for *causal inference* is formalized through Pearl's *do*-operator denoted by $do(temperature)$ as a notion to distinguish the conditional *observational probability* distribution denoted by $\mathbb{P}(Quality \mid temperature)$ from the conditional *interventional probability* distribution denoted by $\mathbb{P}(Quality \mid do(temperature))$. Hence, we distinguish between the observation probability distribution given that we see $Temperature = temperature$ and the interventional probability distribution given that we manipulate $Temperature = temperature$ [130]. Under further identifiability constraints, the CGM, together with the *do*-operator, enables estimating causal effects by examining interventional probability distributions from purely observational data [127].

For example, when examining the causal effect of $Temperature$ on $Quality$, estimating the linear relationship between $Temperature$ and $Qualtiy$ within the conditional observational distribution shows that increasing temperature may reduce the quality of the weld seam; see Fig. 1.7 (a) on page 9. Note that this contradicts the contrary relationship defined in the SCM; see Eq. (1.4) on page 4. On the contrary, the *do*-operator allows resolving this paradoxical behavior. In our example, the *do*-operator is identifiable, as the so-called backdoor criterion allows dissolving the conditional interventional probability distribution as sketched by

$$\mathbb{P}(Quality \mid do(temperature)) = \sum_{steel} \mathbb{P}(Quality \mid steel, temperature)\mathbb{P}(steel).$$

Hence, we adjust for the common confounder $Steel$ to receive the conditional interventional distribution according to the perturbed structure $Steel \rightarrow Quality \leftarrow$



(a) Observational distribution

(b) Interventional distribution

Fig. 1.7: (Simpson's Paradox) Scatterplots depicting the causal effect of $Temperature$ on $Quality$ (smaller better) within our welding process. Estimating the linear relationship between $Temperature$ and $Quality$ shows a negative trend in the observational data; see (a). In contrast, conditioned on the steel grades, this trend is reversed and shows the correct functional relationship as defined in the true underlying SCM Eq. (1.4) (on page 4), which aligns with Simpson's paradox; see (b). The distribution depicted in (b) mimics the conditional interventional distribution, where we adjust for the common confounder $Steel$.

$Temperature$ of an experimental setup. Then, estimating the linear relationship between $Temperature$ and $Qualtiy$ within the conditional interventional observational distribution shows the correct behavior as generated by the underlying SCM defined in Eq. (1.4); see Fig. 1.7 (b). This paradoxical behavior is called Simpson's paradox, which states that a particular trend is reversed when the observational data is conditioned on subgroups [164]. In our welding process, we observe this behavior due to the confounding steel grade, which simultaneously increases the $Temperature$ and decreases $Quality$, see Eq. (1.3) and (1.4). Hence, with the knowledge of the above framework, Simpson's paradox is only paradoxical if we misinterpret the conditional observational probability distribution $\mathbb{P}(Quality \mid temperature)$ as conditional interventional probability distribution $\mathbb{P}(Quality \mid do(temperature))$. In summary, the concepts for causal discovery and causal inference provide a comprehensive framework for data-driven assessment of underlying causal mechanisms that surpass the opportunities of traditional machine learning or deep learning techniques. Therefore, as the 2018 Turing award winner and "Godfather of Deep Learning" Yoshua Bengio stated in a 2019 IEEE Spectrum interview "*Causality is very important for the next steps of progress of machine learning*"[1].

Although this sounds appealing in theory, the above concepts for causal discovery and causal inference contain quite a list of ifs and buts. In particular, if causal discovery results in incorrect causal structures, its interpretation and the application of the *do*-operator are misleading, if not at all, incorrect. Hence, there is ongoing research aiming to provide milder assumptions of causal discovery, e.g., approaches that aim to relax causal sufficiency, faithfulness, or acyclicality, e.g., see [168, 111, 40]. As this is not the focus of this thesis, we refer to [44, 135, 173, 196] for more information on causal discovery in general and recent advances. In real-world scenarios, as we will show below, there are multiple challenges beyond the concepts and assumptions of causal discovery that impede its application in practice. First, given real-world data, selecting an appropriate approach for data preprocessing and causal discovery becomes quite difficult. This is also due to the broad spectrum of different methods and implementations, each having spe-

[1] Bengio Y.: *Yoshua Bengio, Revered Architect of AI, Has Some Ideas About What to Build Next.* In IEEE Spectrum, 2019; https://spectrum.ieee.org/yoshua-bengio-revered-architect-of-ai-has-some-ideas-about-what-to-build-next [Online; accessed September 5th, 2023]

cific assumptions and introducing implementation-specific overheads. Second, the entire construction of the graph relies on correctly detecting (in)dependencies, which require access to a CI oracle. Thus, in practice, small sample sizes, complex underlying causal mechanisms, and mixed discrete-continuous data make this task of selecting the appropriate CI test difficult. These are precisely the challenges we tackle with our research questions, as we elaborate below.

1.3 Challenges and Research Questions

In this section, we examine two omnipresent challenges that impede causal discovery in practice and present our research questions. First, the broad spectrum of implementations in different programming languages and methods with various assumptions impede its evaluation, and application becomes a cumbersome manual task with high setup costs (Section 1.3.1). Second, omnipresent data characteristics of real-world scenarios do not meet the theoretical assumptions of common CI tests, which yields incorrect causal structures (Section 1.3.2).

Note that we focus on these two challenges because they have been painfully apparent in the application of causal discovery in the real-world scenarios of our cooperation partners and have not yet been adequately addressed in related work. For more information on other challenges and recent advances, we refer to [44, 135, 173, 196]

1.3.1 Cumbersome Evaluation and Application of Causal Discovery

In real-world scenarios, determining the correct algorithmic approach and the most suitable implementation for a given observational dataset becomes a tedious manual task. As depicted in Fig. 1.8, applying causal discovery in practice requires the execution of three steps.

First, in *Step 1*, the raw data needs to be preprocessed to generate i.i.d. observations over a set of relevant variables. For example, in an automotive assembly line, data about the production process is stored as raw log data, which may need to be aggregated to

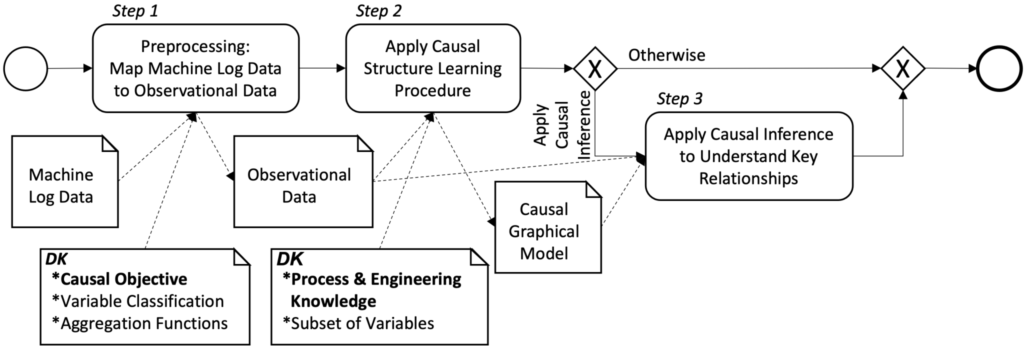


Fig. 1.8: Outline of the causal discovery process from raw machine log data. *Step 1* covers the necessary preprocessing to transform the raw data to observational data, *Step 2* includes selecting an appropriate approach for causal discovery considering the characteristics of the given dataset, and *Step 3* involves evaluating the accuracy the learned causal structures to recognize mistakes in previous steps. This process needs to be repeated until the whole process seems to be flawless and the learned causal mechanisms agree with the partly available domain knowledge (DK).

receive well-defined variables, e.g., see [54]. In this context, mistakes in the preprocessing increase the potential for statistical errors to the detriment of the accuracy of the learned causal structures, e.g., see [104, 29, 139].

Second, *Step 2* requires examining state-of-the-art algorithms with different implementations in different programming languages utilizing different hardware setups. In particular, each algorithm comes with different assumptions on the characteristics of the underlying causal mechanisms and requires selecting various parameters, which all significantly impact the accuracy and the computational complexity [68]. Therefore, the vastly expanding field of research in causal discovery results in a "sheerly unlimited number of methods". Hence, neither is a complete evaluation feasible [37] nor is the computational complexity manageable without an interdisciplinary team incorporating computer scientists, too [5].

Third, in *Step 3*, the learned causal structures need to be validated to recognize mistakes in previous steps. In particular, incorrectly learned structures may hint at errors in the preprocessing or inapplicable assumptions of causal discovery regarding the given dataset, e.g., see [54]. Therefore, this process needs to be repeated until the whole process seems to be flawless and the learned causal mechanisms agree with the partly available domain knowledge. In this context, evaluating is further impeded due to the lack of ground truth in practice and limited benchmark data that reflect characteristics of real-world scenarios [67].

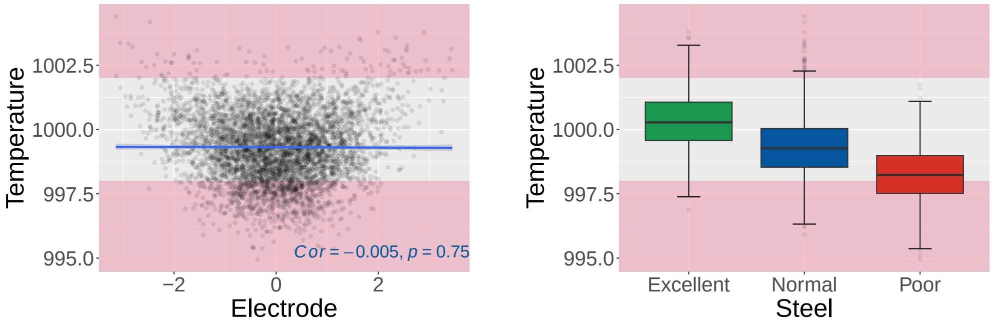
Due to the aforementioned observations, evaluation and application of causal discovery becomes a cumbersome task. Hence, we formulate the following first research question that will be addressed in this thesis:

- **Research Question 1 (RQ1):** *How to support the evaluation of methods for causal discovery and their applicability in practice?*
- **Significance:** The access to tools that enable plugging in existing methods from different libraries into a single system to compare and evaluate the results, also under the prism of improving existing algorithms, is of great importance for researchers and practitioners. Although this is omnipresent in the machine learning domain, where such tools follow a methodology referred to as the common task framework (CTF) [30], providing publicly available benchmarking datasets, a set of competitors, and a scoring referee to compare the competitors, such tools are barely available in the context of causal discovery.

1.3.2 Causal Discovery from Mixed-Discrete Continuous Data

As described in our gentle introduction to causal discovery, see Section 1.2 (page 3), constraint-based causal discovery relies on correctly detecting all (in)dependencies in $P_{\mathbf{V}}$, which require access to a conditional independence (CI) oracle. In this context, commonly used CI tests require variables of the same type or have strong statistical assumptions on the underlying causal mechanisms[2]. In contrast, most real-world scenarios incorporate mixed discrete-continuous data, i.e., variables may follow a discrete, continuous, or discrete-continuous mixture distribution, and underlying causal mechanisms may follow complex nonlinear relationships [67, 69]. For example, in our exemplary welding process, observational data incorporates mixtures of discrete steel grades $Steel$ and continuous temperature measurements $Temperature$ as well as complex non-linear relationships, e.g., for $Electrode \to Temperature$, see Fig. 1.9 on page 12. As depicted in

[2] Note that recent advances in score-based methods for causal discovery allow for mixed discrete-continuous data, e.g., see [64], but are not considered in this thesis as we focus on constraint-based causal discovery.



(a) *Electrode → Temperature* **(b)** *Steel → Temperature*

Fig. 1.9: (Mixed discrete-continuous data) Scatterplot and boxplot depicting the mixed discrete-continuous dependence characteristics in $P_{\mathbf{V}}$. As depicted in (a), *Electrode → Temperature* follows a quadratic relationship such that Pearson's correlation coefficient vanishes, i.e., there is no linear trend, and the correlation is close to zero ($Cor = -0.005$), which implies an incorrect but statistically significant independence statement ($p = 0.75$). As depicted in (b), *Steel → Temperature* shows a mixed discrete-continuous relationship, i.e., poorer steel grades relate to lower, more critical temperatures.

Fig. 1.9 (a) *Electrode → Temperature*, the causal relationship is quadratic as defined in the SCM Eq. (1.3). Hence, the present observational data requires that an appropriate CI oracle must capture both conditional (in)dependencies. In practice, the true statistical properties are mostly unknown such that inadequate assumptions, e.g., of parametric CI tests, yield incorrect learned causal structures [168]. For example, the well-known partial Pearson's correlation-based CI test via Fisher's Z transformation assumes that $P_{\mathbf{V}}$ is multivariate Gaussian [4, 75]. Hence, the underlying causal mechanisms are assumed to be linear, and edges are deleted if the correlation vanishes. But as depicted in Fig. 1.9 (a), Pearson's correlation coefficient vanishes if the causal mechanisms are quadratic, i.e., there is no linear trend, and the correlation (Cor) is close to zero, which implies a statistically significant ($p = 0.75$) independence statement. This becomes even more complex when examining mixed discrete-continuous causal relationships. As depicted in Fig. 1.9 (b) *Steel → Temperature*, the discrete steel grades causally influence the mixed discrete-continuous temperature measurements of our welding process, i.e., poorer steel grades relate to lower, more critical temperatures. Similarly, commonly used CI tests postulate an underlying parametric functional model, which allows for a regression-based characterization of CI. For example, well-known likelihood ratio tests assume conditional Gaussianity (CG) [2, 157] or use multinomial logistic regression models [180]. Hence, similar to Pearson's correlation-based CI test, these require that the postulated parametric models hold, which may yield invalid CI decisions if assumptions are inaccurate, too [168]. Therefore, mixed discrete-continuous data is often transformed to be either discrete or continuous to use standard tests to the detriment of the accuracy of the learned causal structures. In particular, discretization comes with an information loss such that non-linear causal relationships may not be detectable [104, 29, 139].

Due to the aforementioned observations, selecting an appropriate CI test for causal discovery is impeded by the omnipresence of mixed discrete-continuous data. Hence, we formulate the following second research question that will be addressed in this thesis:

- **Research Question 2 (RQ2):** *How to weaken the assumptions of constraint-based causal discovery on data characteristics?*

- **Significance:** As reviewed by Li and Fan [94], there exist a wide range non-parametric CI tests to weaken assumptions in continuous data, e.g., kernel-based approaches, such as `KCIT` [195], or k-nearest neighbors (kNN)-based CI tests, such as `CMIknn` by Runge [148]. In contrast, non-parametric CI tests for mixed discrete-continuous data are barely available and hardly suitable for constraint-based causal discovery. For example, minimum description length (MDL)-based CI tests suffer from their worst-case computational complexity and weaknesses regarding low sample sizes.

1.4 Contributions

In this thesis, we provide a comprehensive answer to the research questions **RQ1** and **RQ2**, which arise from omnipresent challenges when applying causal discovery in practice; see Section 1.3 (page 10).

As depicted in Fig. 1.10, we contribute threefold. First, we tackle **RQ1** and provide tooling to support the evaluation and application of causal discovery.

- **Contribution 1 (C1): Tooling for Causal Discovery**
- **Overview:** We contribute a platform-independent modular pipeline for causal discovery, called `MPCSL` and a ground truth framework for synthetic data generation, called `MANM-CS`, that provide comprehensive evaluation opportunities, e.g., to examine the accuracy of causal discovery methods in case of inappropriate assumptions. For a detailed description of the contributions and information on the related papers, see Section 1.4.1.

Second, we tackle **RQ2** and provide a non-parametric CI test that captures CI characteristics in mixed discrete-continuous data under mild assumptions.

- **Contribution 2 (C2): Non-parametric CI Testing**
- **Overview:** We contribute a non-parametric CI test leveraging k-nearest neighbors methods and prove its statistical validity and power in mixed discrete-continuous data. For a detailed description of the contributions and information on the related papers, see Section 1.4.2.

Third, we validate **C1** and **C1** in practice.

- **Contribution 3 (C3): Validation in Real-World Use Cases**
- **Overview:** To demonstrate the applicability and show the opportunities in practice, we validate our contributions to improve causal discovery in several real-world discrete manufacturing use cases, see Section 1.4.3.

Furthermore, we sketch complementary contributions, e.g., made in the context of hardware acceleration to speed up causal discovery, see Section 1.4.4.

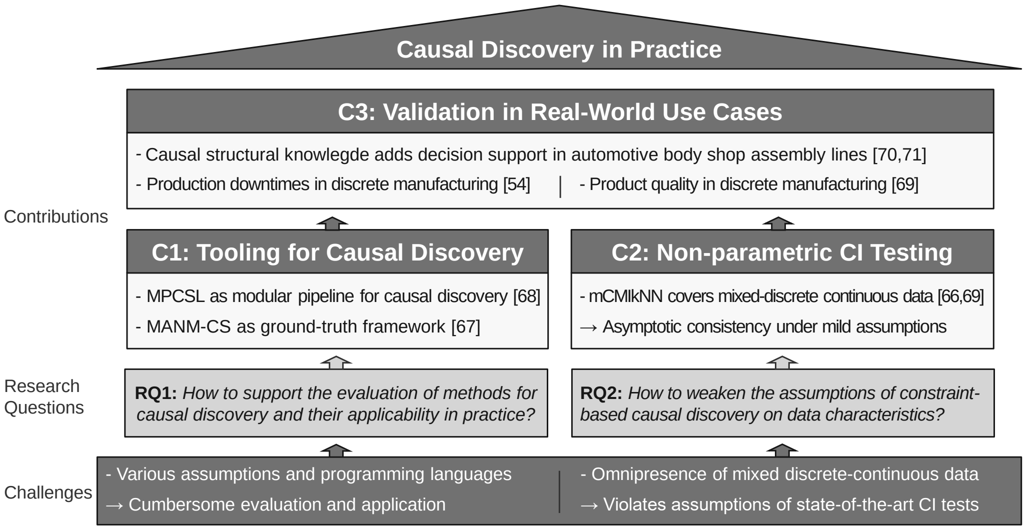

Causal Discovery in Practice
C3: Validation in Real-World Use Cases
- Causal structural knowlegde adds decision support in automotive body shop assembly lines [70,71]
- Production downtimes in discrete manufacturing [54]
- Product quality in discrete manufacturing [69]
Contributions
C1: Tooling for Causal Discovery
- MPCSL as modular pipeline for causal discovery [68]
- MANM-CS as ground-truth framework [67]
C2: Non-parametric CI Testing
- mCMIkNN covers mixed-discrete continuous data [66,69]
→ Asymptotic consistency under mild assumptions
Research Questions
RQ1: How to support the evaluation of methods for causal discovery and their applicability in practice?
RQ2: How to weaken the assumptions of constraint-based causal discovery on data characteristics?
Challenges
- Various assumptions and programming languages
→ Cumbersome evaluation and application
- Omnipresence of mixed discrete-continuous data
→ Violates assumptions of state-of-the-art CI tests

Fig. 1.10: Schematic overview of our contributions **C1**, **C2**, and **C3** (top) to answer the research questions **RQ1** and **RQ2** (center), which arise from omnipresent challenges of causal discovery in practice (bottom).

1.4.1 C1: Tooling for Causal Discovery

With contribution **C1**, we tackle the research question **RQ1** "*How to support the evaluation of methods for causal discovery and their applicability in practice?*". To provide a comprehensive answer, we examine the requirements for tooling, which should enable us to plug existing methods from different libraries into a single system to compare and evaluate the learned causal structures, also under the prism of improving existing algorithms in the context of mixed discrete-continuous data.

To meet these requirements, we propose an architectural blueprint of a pipeline for causal discovery and our reference implementation `MPCSL` that addresses requirements towards platform independence and modularity while ensuring the comparability and reproducibility of experiments [68][3]. In this context, we demonstrate the capabilities of `MPCSL` within a case study, where we evaluate existing implementations of the well-known PC algorithm concerning their runtime performance characteristics. Further, we introduce the mixed additive noise model (MANM) that provides a ground truth model for generating observational data following various distribution models and present our ground truth framework `MANM-CS` [67]. In this context, we demonstrate the usability of `MANM-CS` compared to well-known benchmark data sets and in a simple benchmarking experiment on the accuracy of causal discovery from mixed discrete-continuous data.

The material of contribution **C1** has previously been published in the following peer-reviewed papers:

[68][3] Huegle, J.; Hagedorn, C.; Perscheid, M.; Plattner, H.: *MPCSL - A Modular Pipeline for Causal Structure Learning*. In *Proceedings of the Conference on Knowledge Discovery and Data Mining (KDD)*. 2021, pp. 3068–3076.

[67] Huegle, J.; Hagedorn, C.; Böhme, L.; Pörschke, M.; Umland, J.; Schlosser, R.: *MANM-CS: Data Generation for Benchmarking Causal Structure Learning from Mixed Discrete-Continuous and Nonlinear Data*. In *Neural Information Processing Systems (NeurIPS), Workshop on Causal Inference and Machine Learning: Why Now?* 2021: pp. 1–15.

Contribution: The author of this thesis is the first author of the two publications. The thesis author prepared the original draft of the papers and worked out the necessary concepts as well as the theoretical basis. The implementation was joint work with Christopher Hagedorn, who contributed several ideas and supported the implementation process, which was carried out by all authors and students involved in the respective student projects. The coauthors improved the paper's material and its presentation. A detailed examination of the author's contribution is provided at the beginning of the corresponding chapters Chapter 3 (page 29) and Chapter 4 (page 43).

1.4.2 C2: Non-Parametric CI Testing

With contribution **C2**, we tackle the research question **RQ2** "*How to weaken the assumptions of constraint-based causal discovery on data characteristics?*". To provide a comprehensive answer, we examine the requirements on conditional independence (CI) testing when applying causal discovery in practice. In particular, constraint-based methods require CI tests that yield accurate CI decisions in mixed discrete-continuous data and are computationally feasible as they are applied hundreds of times.

[3] Equal contribution of Johannes Huegle and Christopher Hagedorn.

To meet these requirements, propose `mCMIkNN`, non-parametric CI test that builds upon a k-nearest neighbors (kNN)-based local conditional permutation scheme and a kNN-based conditional mutual information (CMI) estimator as a test statistic [69], which was first sketched in [66]. We provide theoretical results on the CItest's validity and power [69]. In particular, we prove that `mCMIkNN` can control type I and type II errors. Further, we show that `mCMIkNN` allows for consistent estimation of causal structures when used in constraint-based causal discovery [69]. An extensive evaluation on synthetic data shows that `mCMIkNN` outperforms state-of-the-art competitors, particularly for low sample sizes [69].

The material of contribution **C2** has previously been published in the following peer-reviewed papers:

[66] Huegle, J.: *An Information-Theoretic Approach on Causal Structure Learning for Heterogeneous Data Characteristics of Real-World Scenarios*. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Doctoral Consortium Track*. 2021, pp. 4891–4892.

[69] Huegle, J.; Hagedorn, C.; Schlosser, R.: *A kNN-Based Non-Parametric Conditional Independence Test for Mixed Data and Application in Causal Discovery*. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), Part I*. 2023, pp. 541 – 558.

Contribution: The author of this thesis is the first author of the two publications and contributed the theoretical basis and main parts of the implementation, and prepared the original draft of the papers. Christopher Hagedorn supported the implementation and evaluation of `mCMIkNN`. The coauthors improved the paper's material and its presentation. A detailed examination of the author's contribution is provided at the beginning of the corresponding chapters Chapter 8 (page 87) and Chapter 9 (page 103).

1.4.3 C3: Validation in Real-World Use Cases

With contribution **C3**, we demonstrate the applicability of **C1** and **C2** in practice. Therefore, we examine constraints and show opportunities in three real-world discrete manufacturing use cases.

In particular, we motivate our contributions by demonstrating how causal structural knowledge adds decision support in monitoring automotive body shop assembly lines [70, 71]. Furthermore, we show that the results provided by **C1** support causal discovery from manufacturing log data to understand unforeseen production downtimes [54][3]. In particular, we showcase challenges and requirements that arise when dealing with raw log data and provide necessary concepts for the transferability of causal discovery to practice. Moreover, in the appendix of the [69][4], we demonstrate that our non-parametric CI test, see **C2**, outperforms common discretization-based approaches for causal discovery in a real-world discrete manufacturing use case. In particular, we show that `mCMIkNN` correctly detects the (in)dependencies and, hence, allows for a data-driven assessment of underlying causal mechanisms of a machinery production process to improve the production quality performance.

[4] Note that [69] is also listed in **C2**, where we contribute the theoretical basis, implementation, and synthetic evaluation of `mCMIkNN`.

The material of contribution **C3** has previously been published in the following peer-reviewed papers:

[70] Huegle, J.; Hagedorn, C.; Uflacker, M.: *How Causal Structural Knowledge Adds Decision Support in Monitoring of Automotive Body Shop Assembly Lines*. In *Proceedings of the International Joint Conference on Artificial Intelligence (ICJAI), Demos*. 2020, pp. 5246–5248.

[71] Huegle, J.; Hagedorn, C.; Uflacker, M.: *Unterstützte Fehlerbehebung durch kausales Strukturwissen in Überwachungssystemen der Automobilfertigung*. In *Software Engineering (SE)*. Gesellschaft für Informatik, 2021, pp. 1–2.

[54][3] Hagedorn, C.; Huegle, J.; Schlosser, R.: *Understanding Unforeseen Production Downtimes in Manufacturing Processes Using Log Data-Driven Causal Reasoning*. In *Journal of Intelligent Manufacturing* 33(7), 2022: pp. 2027–2043.

[69][4] Huegle, J.; Hagedorn, C.; Schlosser, R.: *A kNN-Based Non-Parametric Conditional Independence Test for Mixed Data and Application in Causal Discovery*. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), Part I*. 2023, pp. 541 – 558.

Contribution: The author of this thesis is the first author of [70, 71, 69]. The thesis author prepared the original draft of the papers and worked out the necessary concepts. In [54], the thesis author is the second author with equal contribution as Christopher Hagedorn, who prepared the original draft of the paper. In this context, all cooperation projects with the industry partners are a joint work of the thesis author and Christopher Hagedorn, led by the thesis author. All authors improved the paper's material and its presentation. A detailed examination of the author's contribution is provided at the beginning of the corresponding chapters Chapter 5 (page 57), Chapter 10 (page 117), and Chapter 12 (page 129).

1.4.4 Complementary Contributions

In addition to the main contributions covered within this thesis, the thesis author contributed in the field of hardware acceleration to speed up causal discovery, e.g., in high-dimensional settings. In this context, the concomitant increase in the runtime of causal discovery algorithms hinders their widespread adoption in practice. Therefore, we design efficient execution strategies that leverage the parallel processing power of graphics processing units (GPUs). In particular, we derive GPU-accelerated variants of the PC algorithm considering different CI tests chosen according to the observational data characteristics and approaches to scaling our GPU-based variants beyond a single GPU's memory capacity. For more information, we refer to the doctoral thesis of Christopher Hagedorn [51].

The material of our complementary contributions to hardware acceleration for causal discovery has previously been published at international conferences, workshops, and technical reports.

[12] Braun, T.; Hurdelhey, B.; Meier, D.; Tsayun, P.; Hagedorn, C.; Huegle, J.; Schlosser, R.: *GPUCSL: GPU-Based Library for Causal Structure Learning*. In *Proceedings of the International Conference on Data Mining (ICDM), Open Project Forum*. 2022, pp. 1236–1239.

[55] Hagedorn, C.; Lange, C.; Huegle, J.; Schlosser, R.: *GPU Acceleration for Information-Theoretic Constraint-Based Causal Discovery*. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), Workshop on Causal Discovery*. 2022, pp. 30–60.

[155] Schmidt, C.; Huegle, J.; Horschig, S.; Uflacker, M.: *Strategies for an Improved GPU-Accelerated Skeleton Discovery for Gaussian Distribution Models*. In *Proceedings of the 2018 HPI Future SOC Lab, Technical Reports*. 2022, pp. 187–197.

[53] Hagedorn, C.; Huegle, J.: *GPU-Accelerated Constraint-Based Causal Structure Learning for Discrete Data*. In *Proceedings of the International Conference on Data Mining (SDM)*. 2021, pp. 37–45.

[52] Hagedorn, C.; Huegle, J.: *Constraint-Based Causal Structure Learning in Multi-GPU Environments*. In *Proceedings of the Lernen. Wissen. Daten. Analysen. (LWDA), Workshop of Fachgruppe Knowledge Discovery and Machine Learning (KDML)*. 2021, pp. 106–118.

[152] Schmidt, C.; Huegle, J.: *Towards a GPU-Accelerated Causal Inference*. In *Proceedings of the 2017 HPI Future SOC Lab, Technical Reports*. 2020, pp. 187–194.

[154] Schmidt, C.; Huegle, J.; Horschig, S.; Uflacker, M.: *Out-of-Core GPU-Accelerated Causal Structure Learning*. In *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*. 2019, pp. 89–104.

[153] Schmidt, C.; Huegle, J.; Bode, P.; Uflacker, M.: *Load-Balanced Parallel Constraint-Based Causal Structure Learning on Multi-Core Systems for High-Dimensional Data*. In *Proceedings of the Conference on Knowledge Discovery and Data Mining (KDD), Workshop on Causal Discovery*. 2019, pp. 59–77.

[156] Schmidt, C.; Huegle, J.; Uflacker, M.: *Order-Independent Constraint-Based Causal Structure Learning for Gaussian Distribution Models Using GPUs*. In *Proceedings of the International Conference on Scientific and Statistical Database Management (SSDBM)* 2018: pp. 19:1–19:10.

Contribution: The first author and main contributor of these publications is Christopher Hagedorn (née Schmidt). The thesis author contributed several ideas, detailed sections regarding the theoretical background of causal graphical models and causal discovery, and improved the papers' material and presentation.

Beyond the topic of causal discovery, the thesis author contributed to several other research projects. This includes the quantitative impact evaluation of data stream processing [60], or research in data-driven agile software process improvement [113, 114]. Moreover, the thesis author contributed to a synthetic simulation framework that enables evaluating self-learning agents for recommerce markets [48].

The material of our complementary contributions to other research projects beyond causal discovery has previously been published at international conferences.

[60] Hesse, G.; Matthies, C.; Glass, K.; Huegle, J.; Uflacker, M.: *Quantitative Impact Evaluation of an Abstraction Layer for Data Stream Processing Systems*. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*. 2019, pp. 1381–1392.

[113] Matthies, C.; Huegle, J.; Dürschmid, T.; Teusner, R.: *Attitudes, Beliefs, and Development Data Concerning Agile Software Development Practices*. In *Proceedings of the International Conference on Software Engineering (ICSE), Software Engineering Education and Training*. 2019, pp. 158–169.

[114] Matthies, C.; Huegle, J.; Dürschmid, T.; Teusner, R.: *Attitudes, Beliefs, and Development Data Concerning Agile Software Development Practices*. In *Software Engineering (SE)*. 2020, pp. 73–74.

[48] Groeneveld, J.; Herrmann, J.; Mollenhauer, N.; Dreessen, L.; Bessin, N.; Schulze Tast, J.; Kastius, A.; Huegle, J.; Schlosser, R.: *Self-Learning Agents for Recommerce Markets*. In *Business and Information Systems Engineering (BISE)*. 2023. *(To Appear)*.

Contribution: The thesis author conducted the statistical analysis in [113, 114], contributed several ideas to all papers, and improved the papers' materials and presentations.

1.5 Outline

In the following, we provide an overview of the structure of this thesis and how it relates to the previously introduced research questions and contributions. As depicted Fig. 1.11, this thesis is structured into three distinct parts.

The first part of this thesis, Part I (page 23), covers the tooling to answer **RQ1** "*How to support the evaluation of methods for causal discovery and their applicability in practice?*". In Chapter 4 (page 43), we introduce `MANM-CS` to provide a framework for generating data characteristics omnipresent in real-world scenarios. In Chapter 3 (page 29), we present a blueprint and a reference implementation, called `MPCSL`, of a modular pipeline for causal discovery. In Chapter 5 (page 57), we demonstrate the process, the challenges, and the opportunities of causal discovery in a real-world discrete manufacturing scenario. In Chapter 6 (page 77), we discuss limitations and future research directions to conclude Part I.

The second part of this thesis, Part II (page 83), introduces our non-parametric CI test to answer **RQ2** "*How to weaken the assumptions of constraint-based causal discovery on data characteristics?*". In Chapter 8 (page 87), we examine the problem of CI testing and related work, provide background on kNN-based CMI estimation, and introduce `mCMIkNN` as well as prove theoretical results. In Chapter 9 (page 103), we empirically evaluate the accuracy of `mCMIkNN` in CI testing and causal discovery compared to state-of-the-art approaches. In Chapter 10 (page 117), we apply `mCMIkNN` in a real-world discrete manufacturing scenario. In Chapter 11 (page 123), we discuss limitations and future research directions to conclude Part II.

The closing part of this thesis, Part III (page 127), showcases how causal structural knowledge adds decision support in the monitoring of automotive body shop assembly lines, see Chapter 12 (page 129). Finally, in Chapter 13 (page 133), we conclude this thesis by summarizing the contributions made to answer our research questions.

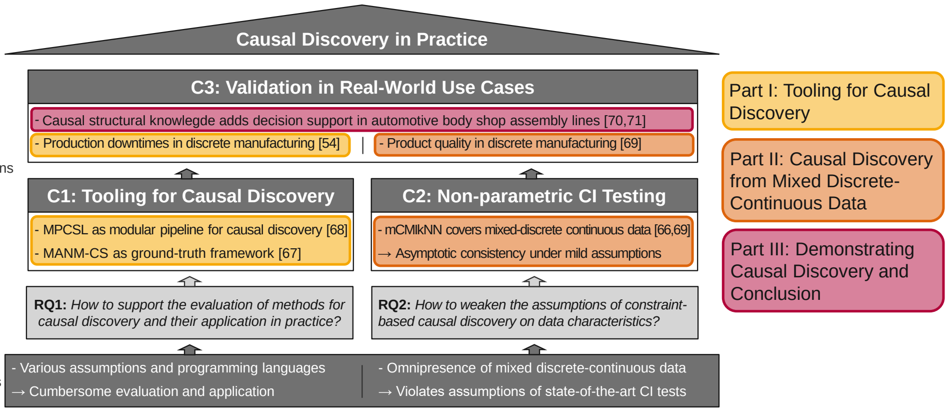

Causal Discovery in Practice
C3: Validation in Real-World Use Cases
- Causal structural knowlegde adds decision support in automotive body shop assembly lines [70,71]
- Production downtimes in discrete manufacturing [54]
- Product quality in discrete manufacturing [69]
Contributions
C1: Tooling for Causal Discovery
- MPCSL as modular pipeline for causal discovery [68]
- MANM-CS as ground-truth framework [67]
C2: Non-parametric CI Testing
- mCMIkNN covers mixed-discrete continuous data [66,69]
→ Asymptotic consistency under mild assumptions
Research Questions
RQ1: How to support the evaluation of methods for causal discovery and their application in practice?
RQ2: How to weaken the assumptions of constraint-based causal discovery on data characteristics?
Challenges
- Various assumptions and programming languages
→ Cumbersome evaluation and application
- Omnipresence of mixed discrete-continuous data
→ Violates assumptions of state-of-the-art CI tests
Part I: Tooling for Causal Discovery
Part II: Causal Discovery from Mixed Discrete-Continuous Data
Part III: Demonstrating Causal Discovery and Conclusion

Fig. 1.11: Schematic overview of our contributions **C1**, **C2**, and **C3** and how they relate to the structure of this thesis. Publications are colored according to the corresponding parts, i.e., Part I (page 23) in yellow, Part II (page 83) in orange, and Part III (page 127) in red.

Part I

Tooling for Causal Discovery

As the knowledge of underlying causal structures is the basis for decision support, causal discovery has received widespread attention. In recent years, the corresponding research addressing challenges of causal discovery in practice has led to a broad spectrum of different methods and implementations, each having specific assumptions and accuracy characteristics or is introducing implementation-specific overhead in the runtime. Hence, methods for causal discovery should be validated within different scenarios, including a varying number of variables or sensitivity of parameters, aiming to understand the method's behaviors in specific edge cases, e.g., when underlying assumptions on the causal relationships are violated. However, considering and evaluating a selection of algorithms or different implementations in different programming languages utilizing different hardware setups becomes a tedious manual task with high setup costs.

Consequently, tooling that enables to plug in existing methods from different libraries into a single system to compare and evaluate the results is substantial support for data scientists in their research efforts. In this context, there is a lack of access to a well-defined ground truth within real-world scenarios to evaluate these methods. In particular, commonly used synthetic benchmarks are limited in their scope as they are either restricted to a "static" low-dimensional data set or do not allow examining mixed discrete-continuous or nonlinear data.

In this part, we tackle **RQ1** "*How to support the evaluation of methods for causal discovery and their applicability in practice?*". To provide a comprehensive answer to this research question, our contributions are threefold. First, we propose `MPCSL`, a pipeline for causal discovery that addresses the requirements towards platform independence and modularity while ensuring the comparability and reproducibility of experiments. Second, we introduce the mixed additive noise model that provides a ground truth framework for generating observational data following various distribution models and present our reference implementation `MANM-CS` to support researchers and practitioners. Third, we demonstrate the process of causal discovery in a real-world discrete manufacturing use case to showcase challenges and requirements and provide concepts for the applicability of causal discovery.

2

Overview on Tooling for Causal Discovery

The knowledge of causal structures is crucial for data scientists in various real-world scenarios. (Section 2.1). In this context, the growing interest and research in methods for causal discovery are confronted with many challenges that impede its application in practice (Section 2.2). Therefore, we contribute by providing the necessary tooling for causal discovery and showcasing its application in a real-world scenario (Section 2.3). Further, we outline Part I of this thesis (Section 2.4).

Contribution: Parts of this chapter have previously been published in the papers [68, 67, 54]. A detailed depiction of the author's contributions is discussed at the beginning of the corresponding chapters Chapter 3, Chapter 4, and Chapter 5, respectively.

2.1 Motivation and Background

The knowledge about the causal structures between the variables of a system is of high interest for data scientists and researchers in a variety of domains [168, 130]. Examples are drug design, where causal graphical models (CGMs) learned from gene expression data depict genetic regulatory relationships [141], or manufacturing, where the knowledge about causal structures supports the root cause analysis of failures within complex production processes, e.g., see [70, 54].

In this context, the growing interest in methods for causal discovery has led to a wide spectrum of scientific publications, each tackling different domain-specific constraints, e.g., statistical considerations to improve the accuracy in non-linear settings [178] or the examination of parallel computing strategies [157, 90, 153] or adoption of accelerator cards [192, 156] to speed-up the learning process. This led to the development of several libraries in different programming languages [151, 157, 75, 90, 179, 86] or standalone implementations [156, 192, 153], each introducing implementation-specific improvements to both the accuracy of the learned causal structures and the runtime of causal discovery.

2.2 Challenges in Practice

Therefore, data scientists of different domains are confronted with the omnipresent challenge of selecting the most appropriate algorithm for causal discovery, given the unique characteristics of their application domain. This may be accompanied by specific hardware and software constraints or unique data characteristics. In these scenarios, determining both the correct algorithmic approach with the corresponding choice of the algorithm's parameters and the most suitable implementation for a given observational data set becomes cumbersome. Even though studies comparing algorithms for causal discovery exist and provide an indication for the best choice, e.g., see [159, 139], they are limited to the algorithms considered and available at the time of writing, as well as the data characteristics utilized in the experimental evaluation. Hence, the evaluation of experiments given an observational data set, considering a selection of state-of-the-art algorithms or different implementations in different programming languages utilizing different hardware setups becomes a tedious manual task with high setup costs.

In real-world scenarios, both missing ground truth and the computational complexity further impede the evaluation of the algorithm's reliability, e.g., see [85, 44]. For example, in genetics, large-scale gene expression data introduces domain-specific constraints with regard to statistical assumptions on the complex, mostly unknown, underlying biological mechanisms on one side and algorithms' performance required for causal discovery in this high-dimensional setting on the other side [37, 5]. Therefore, for biologists aiming to derive gene regulatory networks, the question of practical relevance is the appropriate choice of an existing method for causal discovery for their genetic observational data set specific to their biological research question. As the vastly expanding field of research in methods of causal discovery results in a "sheerly unlimited number of methods", neither is a complete evaluation feasible [37] nor is the computational complexity manageable without an interdisciplinary team that incorporates computer scientists, too [5].

Moreover, data scientists working on advancements of existing algorithms for causal discovery are confronted with the task of considering both naive baseline methods and novel competitive methods in their evaluation. For example in the research field of hardware acceleration in causal discovery, a comprehensive evaluation becomes a tedious manual task with high setup costs as it requires the orchestration of a setup that incorporates not only different programming languages such as `R`, `C++`, or `Python` but also experiments in a heterogeneous hardware setup with multi-core central processing units (CPUs) and GPUs [156].

Glymour et al. summarized the current state as follows: *"There are multiple algorithms available, many of them are poorly tested, some of them are poor implementations of good algorithms, some of them are just plain poor algorithms, all of them have choices of parameters [...], and all of them have conditions on the data distributions and other assumptions under which they will be informative rather than misleading."* [44]. Hence, methods for causal discovery should be validated within different scenarios, including a varying number of variables or sensitivity of parameters, aiming to understand the method's behaviors in specific edge cases, e.g., when underlying assumptions on the causal relationships are violated [88].

2.3 Contributions

In this first part of this thesis, we contribute by providing necessary tool support for the evaluation and application of methods for causal discovery, see **RQ1** in Section 1.3.1 (page 10), in a threefold manner.

- We propose an architectural blueprint of a pipeline for causal discovery, and our reference implementation `MPCSL` to support data scientists in their research on methods for causal discovery.
- We introduce the mixed additive noise model (MANM) that provides a ground truth model for generating observational data following various distribution models and present our reference implementation `MANM-CS`.
- We demonstrate the process of causal discovery in a real-world discrete manufacturing use case to showcase challenges and requirements and provide concepts for the applicability of causal discovery.

2.4 Outline of Part I

The remainder of this part is structured as follows. In Chapter 3 (page 29), we present a blueprint and a reference implementation, called `MPCSL`, of a modular pipeline for causal discovery. In Chapter 4 (page 43), we introduce `MANM-CS` to provide a framework for generating data characteristics omnipresent in real-world scenarios. In Chapter 5 (page 57), we demonstrate the process, challenges, and opportunities of causal discovery in a real-world discrete manufacturing scenario. In Chapter 6 (page 77), we conclude our work and discuss limitations and future research directions.

3

A Modular Pipeline for Causal Discovery

In this chapter, we start by considering requirements for a modular causal discovery pipeline and describe our contributions in more detail (Section 3.1). Further, we consider related work on available tools to support causal discovery (Section 3.2). We provide a blueprint for a pipeline for causal discovery and describe our reference implementation, called `MPCSL` (Section 3.3). Moreover, we showcase its application in a case study on the runtime of causal discovery (Section 3.4). We close this chapter with a conclusion and a discussion on limitations and future work (Section 3.5).

Contribution: Parts of this chapter have previously been published in the paper [68]. The pipeline was developed over multiple student projects. The thesis author worked out the concepts as well as the theoretical basis and ensured the correctness of the applied mathematical concepts. Christopher Hagedorn and the thesis author worked out the experimental evaluation and guided the implementation, which was a joint work together with the involved students. Furthermore, the thesis author prepared the original draft. The coauthors improved the paper's material and its presentation.

3.1 Background: Modular Pipelines for Causal Discovery

Pipelining causal discovery to support the application and evaluation of methods for causal discovery encounters requirements, which follow the common task framework (CTF) (Section 3.1.1). On this basis, we contribute an architectural blueprint and our reference implementation, called `MPCSL` (Section 3.1.2).

3.1.1 Motivation and Requirements on Modular Pipelines

Learning the causal structures between a finite set of variables $\mathbf{V}$ is crucial for data scientists, as it provides data-driven decision support in various application scenarios [130]. For a gentle introduction to causal discovery, see Section 1.2 (page 3) and, for an elaborate background on causal discovery, we refer to [168].

In recent years, the growing interest in research on methods for causal discovery has led to a wide spectrum of independent implementations, each having specific accuracy characteristics and introducing implementation-specific overhead in the runtime. For example, there exist a variety of modifications and extensions of the PC algorithm that can handle a weakened set of assumptions to enable a wider range of applications, e.g., the FCI algorithm [168] allows for causal discovery in the presence of latent confounders, i.e., a violation of causal sufficiency. Hence, considering a selection of algorithms with different assumptions or different implementations in different programming languages utilizing different hardware setups becomes a tedious manual task with high setup costs.

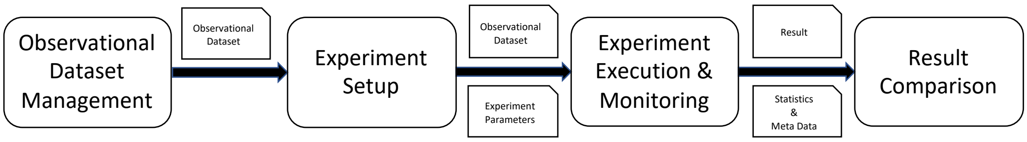


Fig. 3.1: The essential components of the modular pipeline for causal discovery, which are required for a comprehensive benchmarking of different causal discovery algorithm implementations given observational datasets.

Consequently, access to a tool that enables plugging in existing methods from different libraries into a single system to compare and evaluate the results, also under the prism of improving existing algorithms, is of great importance. In the machine learning domain, such tools follow a methodology referred to as the common task framework (CTF) [30], which has the following three components:

- publicly available training datasets;
- a set of competitors;
- and a scoring referee to compare the competitors.

This can be transferred to the process of causal discovery, which can be represented as the pipeline structure depicted in Fig. 3.1. In particular, we introduce the necessary parts of a pipeline that enable incorporating and evaluating existing methods and implementations for causal discovery. Starting with an *Observational Dataset Management* and given an *Experimental Setup*, which enables the selection of both appropriate algorithms and execution environments, the pipeline needs to orchestrate the experiments in a step for *Experiment Execution & Monitoring.* To evaluate the results of causal discovery about both the runtime performance and the accuracy of the results, a *Result Comparison* component completes the pipeline.

The variety of algorithms, different hardware requirements, and constraints of a scientific research setting imply the following requirements for a **platform-independent**, **modular** pipeline for causal discovery that ensures the **comparability and reproducibility** of experiments:

- **Modularity:** The pipeline should be implemented based on the principle of modularity [126] to enable easy adaptation for extensions of the components' feature set, e.g., the addition of new metrics for comparing causal discovery algorithms in different dimensions to support a wide range of evaluation objectives.
- **Platform Independence:** Applying concepts of virtualization and containerization, the pipeline should be designed in a platform-independent manner to support various implementations, independent from the programming language or the hardware requirements, in a single pipeline.
- **Comparability and Reproducibility:** To address the high scientific demand of the research community, all metadata, which contains information regarding the experimental setup, e.g., dataset, algorithms' parameters, or information regarding the underlying hardware setup, should be persisted.

3.1.2 Contribution

Following the previously introduced requirements for a modular causal discovery pipeline, our contributions are threefold.

- We derive a blueprint for a modular causal discovery pipeline to support data scientists in their research on methods for causal discovery.

- We provide a reference implementation, called MPCSL, open accessible to the research community on https://github.com/hpi-epic/mpcsl.
- We demonstrate the MPCSL's applicability in a case study where we determine suitable implementations regarding the runtime of causal discovery for a range of observational data sets.

3.2 Related Work on Tools for Causal Discovery

The challenge of comparing and evaluating a selection of algorithms is omnipresent in the research community and requires an objective evaluation of competitors, as portrayed in the common task framework (CTF) [30]. In the area of causality in general, Dorie et al. [31] implemented the idea of a common task in the "2016 Atlantic Causal Inference Competition", the first large-scale data analysis competition for causal inference from observational data. Though the results explore the relative strengths and weaknesses of methods for causal effect estimation, a sound causal inference requires knowledge about the causal structures of the problem [58]. Hence, causal competitions should evaluate the entire causal inference pipeline incorporating both the examination of methods for causal structure learning and for effect estimation [76].

Several attempts exist to provide tools for the empirical evaluation of methods for causal effect estimation to support an objective comparison between competitors. For example, Lin et al. [98] propose an application programming interface (API) for evaluating causal inference models inspired by the idea of CTF aiming to provide a centralized competition platform for and extended by researchers in causal inference. The researcher can submit new models, datasets, metrics, and parameterizations, which become available to the community and allow for more comparable benchmarks. Shimony et al. [79] present a causal inference benchmarking framework distributed to support a better comparison between methods that estimate causal effects.

In contrast to the task of evaluating methods for causal effect estimation, to the best of our knowledge, there is no common evaluation framework for causal structure learning algorithms across multiple programming languages. Within the R programming language, the package CompareCausalNetworks [57] provides an interface to compare supported causal structure learning algorithms from different classes. In particular, the authors of the package evaluate, PC [168], FCI [168], GES [17], GIES [56], MMHC [182], LINGAM [163] and backshift [147] on simulated graphical networks and use the relations isAncenstor and isParent to compare the learned causal structures across the algorithms. The study provides an overview of the applicability of the methods for researchers, yet it is limited to the considered data characteristics, i.e., not covering any datasets above 100 variables. Furthermore, the package only supports algorithms written in R, not supporting implementations written in other programming languages.

A range of different libraries written in R try to provide a toolbox for causal structure learning, e.g., pcalg [75], or bnlearn [157]. Yet, neither do they support the data scientist with regard to a comparison of runtime performance and accuracy of the results, nor do they cover all existing algorithms. For example, smaller packages in various programming languages implement research in the context of improvements within different statistical settings, e.g., kpc [178] for non-linear causal relationships, or in the context of algorithmic advances for improved runtime performance, e.g., parallelPC [90] or Lock-Free-PC [153] addressing parallel execution strategies, or cupc [192] for execution on accelerator cards. Each library covers a mix of the available approaches and implements particular algorithms in their respective programming language. Hence, a comparison to other implementations for causal discovery becomes a tedious manual task with high setup costs. While there are software tools whose construction follows the pipeline structure of Fig. 3.2 to support a comparison of results, e.g., BayesiaLab [23] or `tetrad` [151], they are limited to the algorithms implemented within the respective software tools.

With our modular pipeline for causal discovery, we provide a framework to incorporate existing libraries and thereby compare implementations across multiple libraries and various programming languages without the need to manually change the experimental setup for each implementation.

3.3 `MPCSL`: A Modular Pipeline for Causal Discovery

In this section, we present a modular pipeline for causal discovery that addresses the requirements on **platform independence** and **modularity** while ensuring the **comparability and reproducibility** of experiments. In particular, we propose an architectural blueprint (Section 3.3.1) and outline our reference implementation called `MPCSL`[5] (Section 3.3.2).

3.3.1 An Architectural Blueprint

In the following, we introduce an architectural blueprint, depicted in Fig. 3.2 on page 33, for a modular pipeline that covers the necessary steps for benchmarking algorithms for causal discovery. It is designed to fulfill the previously mentioned requirements and consists of four main elements. A **Web Application** provides visual support to a data scientist and communicates with the core of the pipeline, the **Backend Service**. A separate service, the **Experiment Execution Service**, is responsible for creating, scheduling, and monitoring dedicated Execution Containers. These **Execution Containers** run specific causal structure learning algorithms and derive the causal graphs based on the observational data stored within the Backend Service.

Web Application: The Web Application provides an interactive front-end that supports the data scientist in the task of creating the experimental setup following the pipeline paradigm depicted in Fig. 3.1 on page 30 (see Fig. 3.2 top). The Web Application communicates with the Backend Service to create the user-defined experimental setup which includes the dataset's source, the method for causal discovery with the corresponding library, and implementation-specific parameters of the algorithm. For an experiment, multiple experiment runs can be executed and thereupon monitored in the Web Application to ensure flawless execution. Once one execution is finished, the Web Application provides mechanisms for the data scientist to explore the determined causal relationships, e.g., visualizing the learned causal graph, examining data distributions, causal dependencies, or causal effect estimation using the *do*-operator [130]. Further, the Web Application visualizes the collected execution statistics, e.g., runtimes of the selected causal discovery implementation, and enables examining predefined accuracy metrics of the learned causal graph, e.g., the false positive rate of edges concerning the expected ground truth graphical model.

Experiment Execution Service: The Experiment Execution Service is responsible for the execution of experiment runs (see Fig. 3.2 left). Upon request by the Backend Service, it creates Execution Containers according to the requirements defined for the experiment, e.g., fitting to the causal discovery implementation and on the requested available hardware. Using independent Execution Containers, which are scheduled on the

[5] https://github.com/hpi-epic/mpcsl

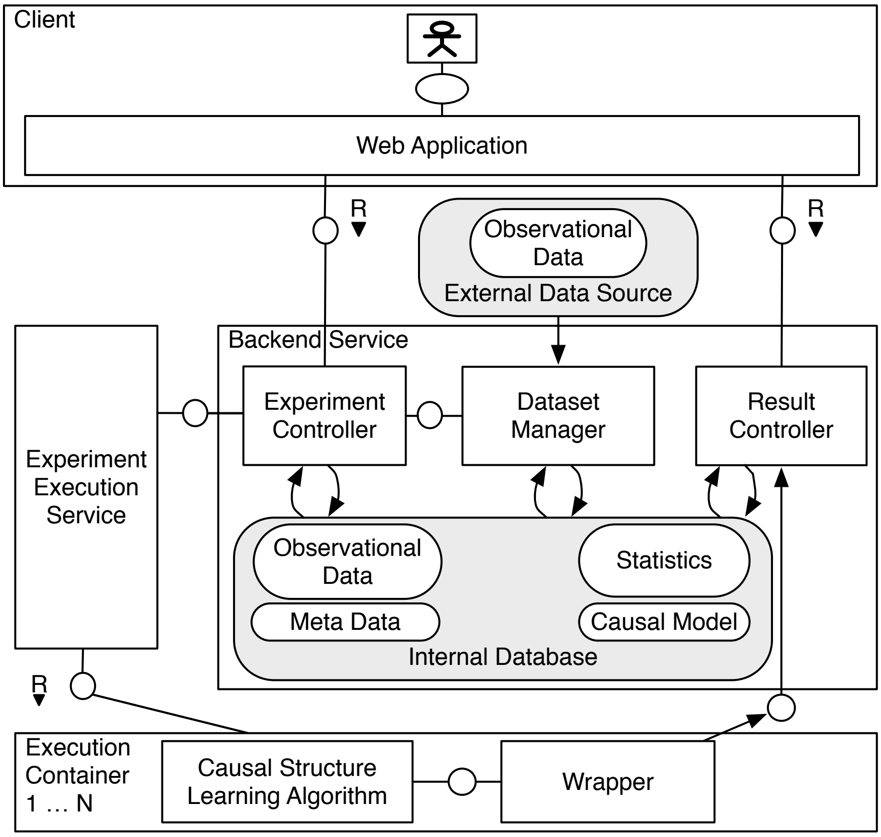


Fig. 3.2: Architectural blueprint of a modular pipeline for causal discovery, which covers the four main elements **Web Application** (top), **Backend Service** (center), **Experiment Execution Service** (left), and **Execution Containers** (bottom).

appropriate hardware, ensures platform independence. The experiment runs are scheduled and, according to the setup, the Experiment Execution Service ensures sequential execution to avoid the influence between different experiments. Further, the experiment runs are monitored and the information is provided to the Backend Service.

Backend Service: The Backend Service handles the requests posted by the Web Application (see Fig. 3.2 center). Therefore, it communicates with the Experiment Execution Service and stores metadata for experiments, e.g., parameter settings and information on datasets to ensure reproducibility and comparability of experiment runs. The Backend Service is composed of a *Dataset Manager*, an *Experiment Controller*, an *Internal Database*, and a *Result Controller*.

The *Dataset Manager* handles all internal and external requests related to the datasets, i.e., creation, deletion, or access. During the creation of a dataset, its metadata, i.e., an identifier, a name and a description of the dataset, the data source, an access method, a validity flag, and a list of its variables, are stored within the Internal Database of the Backend Service. Moreover, a dataset is invalidated in case of critical changes, revoking any new experiment to ensure comparability and reproducibility. The data source points to the location where the observational data is stored, i.e., the Internal

Database or an External Data Source, including access information such as an `SQL` select statement to retrieve the data.

The *Internal Database* stores the above-described metadata on all observational datasets, regardless of their source. Further, it stores the available implementations for causal discovery within the pipeline, the parameter setup of all created experiments, together with the learned causal graph and corresponding statistics collected during execution. In combination with the information on the utilized hardware stored for each experiment run, comparability and reproducibility of results are ensured.

The *Experiment Controller* handles requests from the Web Application concerning experiments and their setup, i.e., registration of observational datasets, creation of experiments, or triggering experiment runs. Registration of observational datasets and obtaining observational datasets for experiment execution are forwarded to the Dataset Manager. The corresponding metadata on the experiments is queried and sent, together with samples from the observational dataset, to the Experiment Execution Service in case an experiment run is triggered.

After execution, the *Result Controller* receives the learned causal graph and collected statistics from the experiment runs through the Wrapper available within the Execution Containers. Further, the Execution Container enables an evaluation of the experiment results through the calculation of accuracy metrics concerning the learned causal graph in comparison to the ground truth stored in the Internal Database. Note that the modular extensibility ensures the incorporation of more advanced accuracy metrics beyond false-positive rates, graph edit distances, or the structural Hamming distance (SHD) that may be required within the respective experimental evaluation, e.g., see [182, 16].

Execution Containers: The Execution Containers are orchestrated by the Backend Service and provide a virtualized environment for the execution of algorithms for causal discovery (see Fig. 3.2 bottom). Thus, appropriate execution environments according to the requirements of the implementations are provided. In addition, the container requires a dedicated *Wrapper* as an interface to communicate with the Backend Service. It provides functionality to receive the dataset and start the execution of the algorithm with the provided parameters. Further, it collects statistics during execution and, upon successful execution, sends the results to the Backend Service.

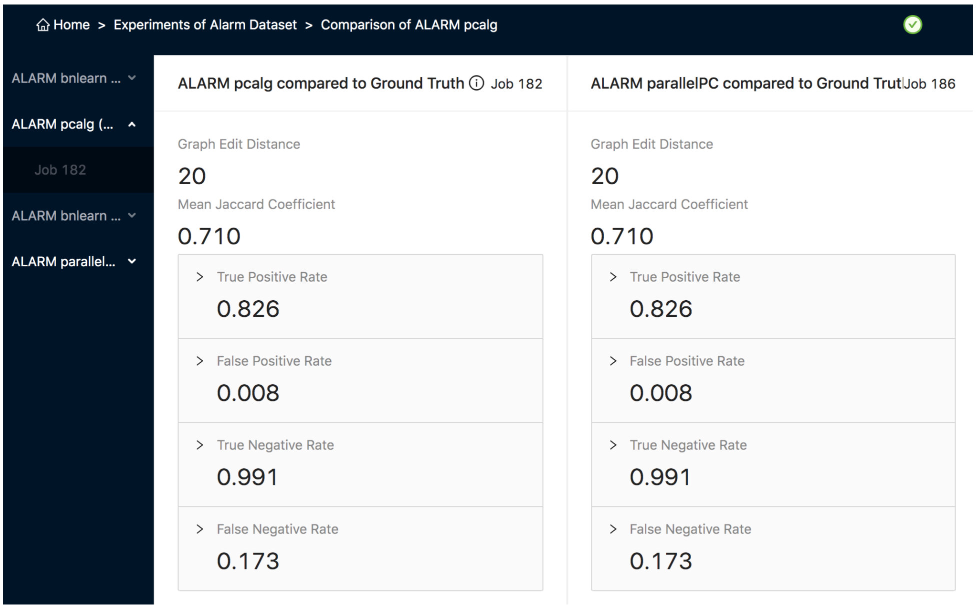
Upon request from the Web Application, the Result Controller provides the learned causal graph, stored in the Internal Database, for a comprehensive exploration and evaluation of the experimental results. The Web Application enables examining the statistics and calculated metrics, e.g., the SHD, in case that ground truth is available for the observational dataset. Moreover, if the ground truth is not available, the provided sampling opportunities from the observational dataset enable a robustness examination, see [85].

In summary, the presented architectural blueprint enables an extensive comparison of implementations for causal discovery with regard to the runtime and accuracy of the learned causal structures.

3.3.2 `MPCSL`: A Reference Implementation

Based on the previously introduced architectural blueprint that covers the requirements on the pipeline for causal discovery, we provide a reference implementation called `MPCSL`[5].

The **Web Application** of `MPCSL` is written in `JavaScript` using `React`, `Redux`, and `D3` libraries. It provides an interface to interact with the different components of the pipeline necessary for an experimental evaluation. It includes components to manage the experiments, to compare and evaluate individual experiment runs, see Fig. 3.3 on page 35, and to explore the learned causal graphs, see Fig. 3.4 on page 35.



ALARM pcalg compared to Ground Truth Job 182

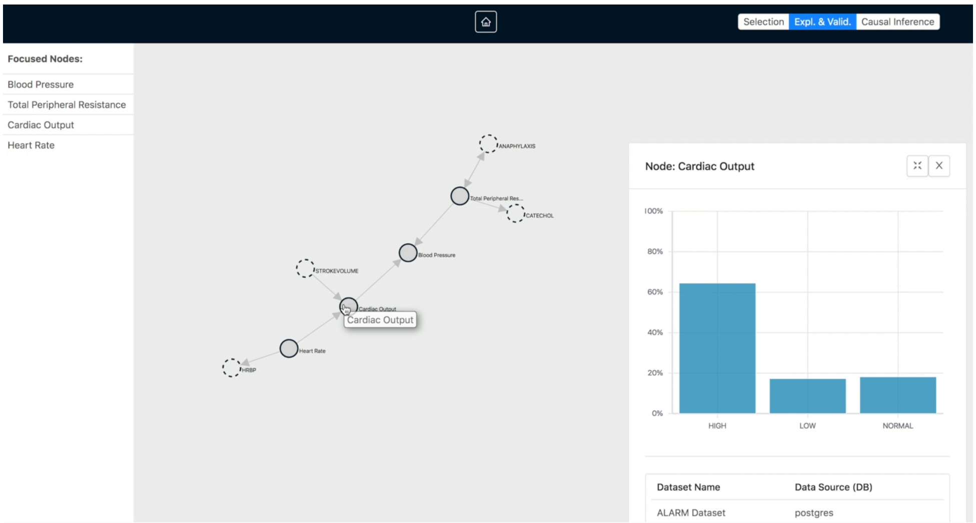
Graph Edit Distance
20
Mean Jaccard Coefficient
0.710
True Positive Rate
0.826
False Positive Rate
0.008
True Negative Rate
0.991
False Negative Rate
0.173

ALARM parallelPC compared to Ground Trut Job 186

Graph Edit Distance
20
Mean Jaccard Coefficient
0.710
True Positive Rate
0.826
False Positive Rate
0.008
True Negative Rate
0.991
False Negative Rate
0.173

Fig. 3.3: The *Validation View* of MPCSL's **Web Application** enables examining the accuracy metrics to compare different experiments, e.g., the graph edit distance or the mean Jaccard coefficient of identical causal graphs learned with pcalg (left) and parallelPC (right) which demonstrate the statistical accordance of the two different implementations.

For example, the *Validation View* enables examining the accuracy of the learned causal structures compared to the ground trough. As depicted in Fig. 3.3, it displays different accuracy metrics such as the graph edit distance or the mean Jaccard coefficient for accuracy evaluation and allows comparing different experiment runs.



Dataset Name	Data Source (DB)
ALARM Dataset	postgres

Fig. 3.4: The *Causal Graph Explorer* of MPCSL's **Web Application** enables examining the learned causal structures and corresponding data distributions.

Moreover, the *Causal Graph Explorer* presents the learned causal structures between variables, using the well-known causal graph representation of edges and nodes as introduced in Section 1.2 (page 3). As depicted in Fig. 3.4, it displays the causal graph and the original data distribution for each variable. Further, it supports functionality to estimate the causal effects of interventions following the *do*-operator paradigm.

The **Backend Service** of `MPSCL` is written in `Python`, using the `Flask` framework [47] to communicate with the *Web Application* through a `RESTful` API. In our reference implementation, the *Dataset Manager* uses SQL to communicate with the *Internal Database*, an instance of a `PostgreSQL` [124] database, and to communicate to external database management systems. We use the `Python SQL` toolkit and the object-relational mapper `SQLAlchemy` [140] to realize the communication and restrict the external data sources to those supported by `SQLAlchemy`.

The **Experiment Execution Service** uses the Python docker package to create individual docker containers for the execution of experiment runs with the corresponding requirements of the experiment.

The **Execution Containers** are based on `Docker` [116]. Currently, `MPCSL` provides docker images that support the `R` language, `Python` and that support the `CUDA` framework, when running on systems that include NVIDIA graphics processing unit (GPU) hardware. Through the `R` execution environment, we support the packages `pcalg` [75], `ParallelPC` [90] and `bnlearn` [157]. With the `CUDA` execution environment, we added support for GPU-accelerated implementations [53, 156, 192].

For each experiment run, `MPCSL` collects and presents a selection of statistics, e.g., the runtime or the number of learned edges. Moreover, the *Result Controller* provides the common accuracy metrics, such as the SHD or false positive rates of the learned causal model with respect to edges within the ground truth CGM. The examination of accuracy metrics, such as the graph edit distance or mean Jaccard coefficient, for differently learned causal structures enables a direct comparison of experiments on the same dataset, compare Fig. 3.3, and, in combination with sampling, provides the basis for a robustness examination [85].

All implemented `MPCSL` services run in separate containers. The implementation uses `Kubernetes` to orchestrate the containers and execute them on different physical host systems, depending on the requirements of the causal discovery implementation.

3.4 A Case Study on Runtime of Causal Discovery

In this section, we showcase the capabilities of `MPCSL` in a case study on the runtime of causal discovery. In particular, after motivating our experimental evaluation (Section 3.4.1) and describing our experimental setup (Section 3.4.2), we evaluate the runtime of different PC-stable implementations that are executed single-threaded (Section 3.4.3), parallel (Section 3.4.4), and on graphics processing units (GPUs) (Section 3.4.5).

3.4.1 Motivation

As previously introduced, data scientists trying to apply methods for causal discovery in real-world scenarios are confronted with a wide spectrum of different implementations. There is work that evaluates the different causal discovery approaches with respect to the accuracy of learned causal models, e.g., see [159, 57]. While this may support a data scientist in selecting the appropriate approach, there is, to the best of our knowledge, no work that evaluates the runtime of different implementations on datasets with different data distributions. Especially medical scenarios, such as genetics, require fast implementations that allow for causal discovery in these high-dimensional settings [75, 5]. As different implementations are written in different programming languages and utilize different hardware setups, an evaluation about runtime becomes a tedious manual task with high setup costs.

In the following case study, we illustrate the capabilities of `MPCSL` when comparing a selection of constraint-based causal discovery implementations with respect to their runtime. We choose constraint-based causal discovery for the following reasons. First, it is widely used in real-world scenarios [70, 90], making it an interesting case for practitioners. Second, it provides flexibility to adjust to different data distributions and dependencies by changing the utilized conditional independence (CI) test accordingly. Third, constraint-based causal discovery implementations are available in many different programming languages, libraries, and packages. In particular, we focus on the well-known PC algorithm [168], especially its order-independent version PC-stable [20]. Existing implementations with varying hardware acceleration of the PC-stable [75, 157, 53, 90, 192, 153, 156, 151, 86] are based upon the same algorithm, hence, should have equal accuracy for identically selected parameters, e.g., see Fig. 3.3. Yet, specific implementation details impact the runtime, making it an interesting case for an experimental evaluation.

Our case study consists of three separate experiments on datasets with different data distributions. Further, the datasets differ in the number of variables $N = |\mathbf{V}|$ and the number of samples n. Both characteristics impact the runtime of the PC-stable algorithm.

First, we conduct multiple experiment runs examining single-threaded runtimes of two different causal discovery implementations to provide an indication of the suitability in the context of given data distribution with the corresponding CI test, provided by the respective library; see Section 3.4.3. In the second and third experiments, we examine the potential of modern hardware to reduce the runtime of causal discovery. Therefore, we consider the speed-up of causal discovery implementations on a multi-core server, see Section 3.4.4, and investigate the runtime of a GPU-accelerated implementation, see Section 3.4.5.

3.4.2 Experimental Setup

In the following, we provide an overview of the observational datasets, the hardware setup, and the causal discovery implementations supported by `MPCSL` for the experiments of our case study. During the setup of `MPCSL`, we added the datasets as described below. We use the Web Application to parameterize the experiments for each causal discovery implementation. The experiment execution on the chosen hardware setup is orchestrated by `MPCSL`'s Experiment Execution Service. After execution of multiple experiment runs, the requested metric, for our case study, the median runtime, is presented in the Web Application.

Data: The characteristics of the observational datasets loaded into `MPCSL` for the experimental evaluation are provided in Table 3.1.

The datasets are either real-world gene expression datasets following a multivariate Gaussian distribution, used in existing evaluations [90, 192, 156] (Table 3.1 top), or generated datasets according to the well-known reference Bayesian networks from the `bnlearn` [157] network repository (Table 3.1 bottom). The datasets contain different numbers of variables, ranging from, e.g., $N = 24$ in `MEHRA` up to $N = 5\,361$ in `S.CEREVISIAE`. Thus, the datasets cover both low-dimensional datasets, which we define as $N < 500$, and high-dimensional datasets, i.e., $N \geq 500$. Furthermore, the datasets incorporate different data distributions, for instance, multivariate Gaussian distributed data, called *Gaussian*, multinomial distributed data, called *Discrete*, and conditional Gaussian distributed data, called *Mixed*.

Table 3.1: Characteristics of gene expression datasets (top) and datasets from the bnlearn repository (bottom) with different numbers of variables N and numbers of samples n.

Dataset	Distribution	N	n	Dimension
NCI-60	*Gaussian*	1 190	47	High
MCC	*Gaussian*	1 380	88	High
BR51	*Gaussian*	1 592	50	High
DREAM5-INSILICO	*Gaussian*	1 643	850	High
S.AUREUS	*Gaussian*	2 810	160	High
S.CEREVISIAE	*Gaussian*	5 361	63	High
ARTH150	*Gaussian*	107	20 000	Low
ALARM	*Discrete*	37	10 000	Low
ANDES	*Discrete*	223	20 000	Low
LINK	*Discrete*	724	20 000	High
MUNIN	*Discrete*	1 041	20 000	High
MEHRA	*Mixed*	24	20 000	Low

Hardware Setup: All experiment runs are executed in a Kubernetes cluster consisting of a virtual machine, running the MPCSL Experiment Execution Service, the MPCSL Backend Service, including the Internal Database and an enterprise-grade server with 2 Intel® Xeon® Gold 6148 CPU with 20 cores each. The server is equipped with 1.5 TB of RAM, allowing to keep all data in memory during the execution of the experiment runs, and has a 4 NVIDIA V100 card, with 32 GB of high bandwidth memory. The Execution Containers are run on the server. Furthermore, datasets are fetched from an external database management system running on a separate system. During the runtime measurements, only a single container is executed at a time and no other operations are executed on the server. Measurements are conducted within the MPCSL *Wrapper*, measuring the algorithm's function call, e.g., for R library bnlearn [157] using the difftime of the Sys.time() prior to and after the call of pc.stable. If not stated differently, we repeat each experiment run at least 10 times and present the median runtime. Further, we set the tuning parameter α to 0.01, which is common in application [20].

Employed Causal Discovery Implementations: We compare implementations of the PC-stable [20] from the well-known libraries pcalg [75] and bnlearn [157], as well as, from the libraries Parallelpc [90] and cupc [192], which allow for a comparison of different parallel execution strategies, and hardware acceleration in GPU-accelerated environments. The four implementations support *Gaussian* data using a Fisher's Z test. For *Discrete* data bnlearn uses Pearson's X^2 test, pcalg and Parallelpc use the very similar G^2 test and cupc has no test implemented. *Mixed* data is only supported by bnlearn that incorporates a mutual information-based CI test for conditional Gaussian settings, provided through the mi-cg implementation. Parallelpc is written entirely in R and uses the R library parallel for parallel execution. Bnlearn is written in R and C, using C to provide efficient implementations for most functions. It also uses the R library parallel for parallel execution. Pcalg is written in R and implements the test for *Gaussian* data in C++. For this particular case, it uses openMP [26] for parallel execution. In contrast to the previous libraries, cupc targets a heterogeneous system with an NVIDIA GPU. It provides an R Interface, yet the core algorithm is written using the CUDA framework [122] to utilize the parallel computing capabilities of a GPU.

Table 3.2: Median runtimes in seconds of PC-stable in single-threaded execution. Comparing implementations from libraries `pcalg` and `bnlearn` on benchmark datasets (see Table 3.1). Note, for algorithms exceeding the runtime of a day, we executed a single run or terminated after five days.

Dataset	`pcalg`	`bnlearn`
`ARTH150`	248.7	7 617.0
`NCI-60`	6.7	45 251.1
`MCC`	21.5	83 886.0
`BR51`	32.3	171 629.7
`DREAM5-INSILICO`	4 257.0	360 572.2
`S.AUREUS`	180.0	> 432 000.0
`S.CEREVISIAE`	108.2	> 432 000.0
`ALARM`	30.1	0.5
`ANDES`	2 824.0	108.8
`LINK`	324 873.4	3 827.2
`MUNIN`	> 432 000.0	6 753.0
`MEHRA`	-	11.9

3.4.3 Single-Threaded Runtime

In the following, we measure the single-threaded runtime of the PC-stable implementation from `pcalg` and `bnlearn` on all datasets presented in Table 3.1. The measured runtimes are shown in Table 3.2.

For the *Gaussian* datasets, our measurements show that the implementation from the `pcalg` library outperforms the `bnlearn` implementation by factors of 30 for low-dimensional datasets, e.g., `ARTH150` and factors of up to 6 753 for high-dimensional datasets. For the datasets `S.AUREUS` and `S.CEREVISIAE`, we terminated the execution of `bnlearn` after 5 days without any result, whereas `pcalg`'s implementation finished after 180 seconds. Both implementations use Fisher's Z test for CI testing and are written in `C` or `C++`. Yet, `pcalg` works on a pre-calculated correlation matrix, whereas within `bnlearn` the correlation is calculated repeatedly for each independence test, which we account for the significant difference in measured runtime. Note, the calculation of the correlation matrix prior to the execution of the PC-stable, for the datasets subject of the study ranges from 0.044 seconds for `NCI-60` to 1 391 seconds for `S.CEREVISIAE`. Thus, it does not impact the total runtime in a significant way.

For the *Discrete* datasets, the implementation from the `bnlearn` package outperforms `pcalg` implementation by factors of up to 25, for low-dimensional datasets, e.g., `ALARM`, to factors of up to 84 for higher-dimensional datasets, e.g., `LINK`. For the dataset `MUNIN`, the execution of the `pcalg` implementation was terminated after 5 days, without any result, compared to `bnlearn`'s implementation, which finished below 2 hours. We account the difference to the fact that the CI test for *Discrete* data in `pcalg` is written in `R`, whereas `bnlearn` uses efficient `C` code.

For the *Mixed* dataset `MEHRA`, we report numbers only for `bnlearn`, as no other library used in the case study supports *Mixed* data.

Based on the measurements, we conclude that it is advisable to use the implementation from the `pcalg` library for *Gaussian* data, whereas for *Discrete* and *Mixed* data the `bnlearn` library should be used.

Table 3.3: Factors of speed-up measured with an increasing number of cores running parallel versions of the PC-stable of different implementations compared to the single-threaded execution for different benchmark datasets (see Table 3.1).

Library	Dataset	Cores			
		5	10	20	40
Parallelpc	ARTH150	3.3	4.1	6.9	7.2
bnlearn	ARTH150	3.3	5.3	8.3	11.7
pcalg	ARTH150	3.0	5.7	8.9	15.5
Parallelpc	ANDES	3.4	6.2	10.9	14.3
bnlearn	ANDES	1.6	1.6	1.6	1.4
bnlearn	LINK	3.4	5.3	7.0	7.6
bnlearn	MUNIN	4.0	6.8	11.3	14.4

3.4.4 Parallel Efficiency on Multi-Core Systems

In the following experiment, we measure the runtime of parallel implementations of the PC-stable algorithm, taken from the libraries `pcalg`, `Parallelpc` and `bnlearn`. In previous studies [158, 90], the benefit of parallel execution to reduce the runtime of constraint-based causal discovery algorithms is shown, yet, to the best of our knowledge, no comparison between library-specific implementations exists. Therefore, we aim to determine differences in the implementations' behavior, when scaling the number of cores. All three implementations follow the framework for parallel constraint-based causal discovery learning [158]. Yet, parallel processing within the implementations of the PC-stable is applied only to the adjacency search, which comprises the CI tests. Other parts of the implementations are executed sequentially, which limits the effect of scaling the number of cores on the measured runtime.

In Table 3.3, we provide our measurements for the *Gaussian* dataset `ARTH150` using all three libraries and for *Discrete* datasets using `Parallelpc` and `bnlearn`, given that `pcalg` does not support parallel execution for *Discrete* data. For all measurements, we observe that the ideal speed-up is not achieved, which is due to the sequential part of the implementations.

For the *Gaussian* dataset `ARTH150`, `Parallelpc` shows the implementations' worst speed-up, achieving only a factor of up to 7.15 on 40 cores compared to a factor 15.51 speed-up of `pcalg`. Except for this particular difference, the behavior is similar for all implementations. Note that the overall runtime remains significantly slower for `bnlearn` compared to `pcalg`, see Table 3.2 on page 39.

Considering *Discrete* datasets, we observe a runtime improvement for `bnlearn` using up to 40 cores for the high-dimensional datasets `LINK` and `MUNIN`. For the dataset `ANDES`, speed-up through the increase in cores is small with factors below 2 and even decreases when running on 40 cores compared to 20 cores. We assume that the sequential part of the execution dominates the processing of the dataset `ANDES` in `bnlearn` and that the adjacency search is already efficient in single-threaded execution. In contrast, processing `ANDES` with `ParallelPC`, we observe a strong speed-up, yet the overall runtime for 40 cores is slower than the single-threaded execution with `bnlearn`. This supports our assumption that `bnlearn` has efficient adjacency search and CI test implementations for *Discrete* data.

Overall, we did not observe a significant difference in the implementations' behavior with regard to scaling the number of cores. Yet, for lower-dimensional datasets, e.g., `ANDES`, efficient implementations, e.g., found in `bnlearn`, cannot benefit from multi-core systems and may even be slowed down.

Table 3.4: Median runtimes in seconds of PC-stable with the adjacency search executed on a GPU using `cupc` or in parallel using 40 CPU cores executed with `pcalg` for different benchmark datasets (see Table 3.1).

Dataset	`cupc`	`pcalg` (40 cores)
`ARTH150`	1.93	16.03
`NCI-60`	1.94	3.18
`MCC`	2.45	4.93
`BR51`	4.48	6.42
`DREAM5-INSILICO`	5.41	168.09
`S.AUREUS`	13.58	35.21
`S.CEREVISIAE`	32.79	55.59

3.4.5 GPU-Acceleration

In a third experiment, we consider a heterogeneous system and investigate the speed-up of a GPU-accelerated implementation, `cupc`, over a CPU-based parallel implementation, `pcalg` running on 40 cores. The `MPCSL` *Experiment Execution Service* ensures that the experiment runs are scheduled on hardware that fulfills the requirements to execute the implementations, e.g., provide an NVIDIA GPU for `cupc`.

In Table 3.4, we report the measured runtimes on all *Gaussian* datasets considered in our case study. Using a GPU to process the datasets leads to additional speed-up for all datasets, ranging from speed-up of factor 1.43 for dataset `BR51` to speed-up of factor 31.07 for dataset `DREAM5-INSILICO`. We observe that the two datasets, `DREAM5-INSILICO` and `ARTH150`, result in higher speed-up when the adjacency search is executed on the GPU. Both datasets have a larger number of observations N, compared to the other datasets, which results in an increased number of higher-order CI tests. For the other five datasets, execution on the GPU leads to a speed-up of less than factor 2. For these datasets, over 94% of the independence tests are either unconditioned or with a conditioning set of size 1, e.g., see [156], for the given tuning parameter $\alpha = 0.01$. We account for the high percentage of low-order independence tests for the small factor of speed-up achievable with GPU-acceleration.

3.5 Discussion

We close this chapter with a summary of our contributions (Section 3.5.1), an examination of limitations (Section 3.5.2), and future work (Section 3.5.3).

3.5.1 Summary

We presented a blueprint and a reference implementation of a modular pipeline for causal discovery called `MPCSL`. The pipeline supports the experimental evaluation of the wide spectrum of implementations of algorithms for causal discovery from data scientists and their application in practice. In particular, the platform manages the observational datasets and parameter setups for experiment runs. Further, it orchestrates the execution of the experiments with given hardware constraints using virtualized environments. Storing the parameterization ensures reproducibility. The collection of runtime statistics and accuracy metrics allows for a comprehensive experimental evaluation in the pipeline.

We presented the capabilities of our reference implementation `MPCSL` that was developed to address the requirements of deployment in scenarios from both a theoretical and practical perspective. In detail, we conduct a case study using `MPCSL` to investigate the runtime performance of implementations of the well-known PC algorithm [168]. In this scenario, the results indicate that for *Gaussian* data, the `pcalg` library provides better runtimes by factors of over $1,000$, whereas for *Discrete* data, the `bnlearn` library outperforms the `pcalg` implementation by factors of up to 84. An investigation of parallel execution on multi-core systems showed similar behavior with an increasing number of cores across the implementations, achieving a speed-up of up to factor 15.5 on 40 cores. Additional speed-up of factors of up to 31 is possible employing GPUs, yet their application is currently limited to *Gaussian* data.

3.5.2 Limitations

As the goal of developing a universal modular pipeline that covers all aspects of causal discovery and causal inference is a neverending story, we restricted `MPCSL`'s capabilities to cover the "basic" set of tools and methods.

Hence, due to the thesis' focus on causal discovery under the common assumptions (see Section 1.2 on page 3), the current implementation of `MPCSL` does not cover algorithms that allow for latent variables, e.g., the FCI algorithm [168], or enable causal discovery when interventional data is partly available, e.g., see [13, 133]. Furthermore, we simplified the capabilities of `MPCSL` for causal inference to the case of discrete variables, as it simplifies application of the *do*-operator (see Chapter 5 on page 57). Therefore, `MPCSL` does not cover more complex concepts for causal inference, e.g., causal strengths [72] or estimating causal effects as average treatment effects (ATE) [129, 168].

Moreover, `MPCSL`'s visualization capabilities are restricted as well. For example, the reference implementation is limited to showing a single causal graph which impedes the comparability of two experiment runs' learned CPDAGs.

Currently, `MPCSL` lacks the opportunity to generate datasets with different data characteristics following a selection of data-generating models such that the pipeline envelopes the whole benchmarking pipeline. Therefore, we developed `MANM-CS`, see Chapter 4, as a ground truth framework for synthetic data generation for causal discovery, but `MANM-CS` is not included in `MPCSL`. Nontheless, generated data from `MANM-CS` can be easily loaded into `MPCSL` as described in Section 3.3.2.

3.5.3 Future Work

While `MPCSL` currently serves as a starting point, the pipeline's modularity with respect to the CTF enables for extensibility regarding the causally insufficient case or interventional data. In this context, we aim to extend the platform to directly provide a larger set of causal discovery implementations to further raise the interest of the research community. We invite the research community to actively participate in the extension of `MPCSL`.

4

Synthetic Data Generation for Causal Discovery

In this chapter, we introduce the mixed additive noise model (MANM) that provides a ground truth framework for generating observational data following various distribution models omnipresent in practice. We start by considering requirements for modeling causal structures and describe our contributions in more detail (Section 4.1). Further, we consider related work on available benchmarking methods of causal discovery (Section 4.2). We introduce the MANM as a benchmarking framework and demonstrate its application in several scenarios (Section 4.3). Furthermore, we present our reference implementation `MANM-CS` (Section 4.4) and its application in a case study for benchmarking causal discovery (Section 4.5). We close this chapter by discussing limitations and future work (Section 4.6).

Contribution: Parts of this chapter have previously been published in the paper [67]. The data generator was developed in a student project. The thesis author worked out the concepts and theoretical basis of the data generator. Hagedorn and the thesis author worked out the experimental evaluation and guided the implementation, which was a joint work by all authors. Furthermore, the thesis author prepared the original draft. The coauthors improved the paper's material and its presentation.

4.1 Background: Benchmarking Methods for Causal Discovery

In this section, we motivate by examining how the growing research on causal discovery encounters requirements concerning well-defined benchmark data, particularly mixed discrete-continuous or nonlinear data (Section 4.1.1) Therefore, we contribute the mixed additive noise model (MANM) to establish a flexible yet well-defined ground truth model, allowing the data generation under various evaluation perspectives (Section 4.1.2).

4.1.1 Motivation and Requirements for Synthetic Data Generation

In the following, the following standard notation as introduced in Section 1.2 is used. The causal structures between a finite set of V random variables $\mathbf{V} = \{V_1, \dots, V_N\}$ are encoded in a causal graphical model (CGM) consisting of a directed acyclic graph (DAG) $\mathcal{G}$, where directed edges $V_j \rightarrow V_i$ depict a direct causal relationship between two respective nodes V_j and V_i, $i, j = 1, \dots, N$, and the joint distribution over the variables $\mathbf{V}$, denoted by $P_{\mathbf{V}}$, e.g., see [130, 168].

Within this framework, causal discovery aims to derive as many of the underlying causal relationships in $\mathcal{G}$ from independent and identically distributed observational data as possible. For a gentle introduction to causal discovery, see Section 1.2 (page 3) and, for an elaborate background on causal discovery, we refer to [168].

While all methods for causal discovery require that several assumptions hold (see Section 1.2 on page 3), observational data of real-world scenarios often violates the constraints made for causal discovery. For example, in practice, it may be impossible to observe all variables to ensure causal sufficiency, i.e., that there are no latent confounding variables [168]. Moreover, real-world data often does not follow a simple functional form but includes nonlinear and mixed discrete-continuous relationships [44, 106]. Therefore, a wide spectrum of scientific publications focus on different extensions to improve the accuracy under weakened constraints, e.g., given latent variables [21, 170], assuming a nonlinear function f within the structural causal model (SCM) [63, 194] or considering causal discovery from mixed discrete-continuous data [2, 64, 161, 180].

As (novel) methods for causal discovery are commonly evaluated against their own synthetic benchmarks and compared within a limited scope, e.g., [139, 165, 182], it is difficult to compare individual methods against each other, particularly, if they may require different assumptions. In this context, Glymour et al. [44] summarized the current state as follows: *"There are multiple algorithms available, many of them are poorly tested [...], all of them have choices of parameters [...], and all of them have conditions on the data distributions and other assumptions under which they will be informative rather than misleading."*. Hence, methods for causal discovery should be validated within different scenarios, including a varying number of variables or sensitivity of parameters, aiming to understand the method's behaviors in specific edge cases, e.g., when underlying assumptions on the causal relationships are violated [88]. Therefore, a thorough evaluation of methods for causal discovery requires the introduction of an easily customizable framework for generating observational data supplemented with precise definitions of underlying causal structures that connect and extend existing ideas, see Section 4.2. In particular, a data-generating model should satisfy the following requirements:

(R1) be formalized as a SCM to ensure interoperability of causal relationships, e.g., concerning causal inference;

(R2) allows for continuous, discrete, and mixed discrete-continuous causal relationships, i.e., to mimic data characteristics in practice;

(R3) be flexible and easily extendable, e.g., to allow for interventional data;

(R4) be implemented as an easy-to-use open access package.

4.1.2 Contribution

In this chapter, we propose the mixed additive noise model (MANM) as a flexible yet easy-to-use synthetic data generation process for benchmarking methods for causal discovery under a wide range of conditions. Our main contributions can be summarized as follows:

- We introduce the MANM as a SCM to model causal structures within various distribution models that incorporate discrete, mixed discrete-continuous, or nonlinear causal relationships, see (R1)-(R3).
- To provide easy access to the research community, we present our reference implementation, called `MANM-CS`, open accessible on https://github.com/hpi-epic/manm-cs, see (R4).
- We demonstrate the usability of `MANM-CS` in comparison to well-known benchmark data sets and in a simple benchmarking experiment on the accuracy of causal discovery from mixed discrete-continuous and nonlinear data.

4.2 Related Work on the Benchmarking of Causal Discovery

Usually, the accuracy of methods for causal discovery is evaluated within different synthetic benchmark scenarios to identify their strengths and weaknesses. As the simulated data is often not publicly available, there is a lack of comprehensive comparison with other state-of-the-art methods [139]. Therefore, it is desired to create a general and flexible framework that can be used to benchmark causal discovery methods, particularly within data with real-world characteristics.

Commonly, data for evaluating methods for causal discovery is generated according to the following approaches:

(I) predefined benchmark data sets supplemented by an expected ground truth;

(II) well-established parameterized benchmark models to generate data; and

(III) flexible models based upon a probabilistic or functional formalization.

In Table 4.1, we recap a selection of the above approaches that have been used for evaluation within work on causal discovery, e.g., [2, 139, 161, 180, 182]. We do not claim completeness but focus on the most well-known and representative data sets or models. In this context, currently used models and data sets of the three approaches come with limitations that restrict the evaluation opportunities.

Predefined benchmark data sets (I) allow a direct comparison given a common and enclosed ground truth model. However, they do not allow for performance comparison concerning a varying complexity or data set size. For example, the "DREAM5 SYSGEN A - In-silico network challenge" [109] is based on simulated gene expression data from [100] restricted to 1 000 variables and sample sizes of $n = 100$, $n = 300$, or $n = 999$. To allow for performance evaluation of large sample properties, models from (II) sample observational data from well-established "static" parameterized models with fixed model complexity. While this allows evaluating and comparing methods for causal discovery within the provided distribution and model assumption, they do not allow for an examination given a varying model complexity. For example, within the mixed case, the well-known conditional Gaussian distributed MEHRA model from [183] is restricted to 24 nodes that incorporate a mixture of 8 discrete and 16 continuous variables. Further, within the discrete case, the ALARM model from [6] fixes the number of possible discrete values each variable can take to the model's assumptions. Therefore, following approach (III), most causal discovery methods are evaluated within their respective scenarios, e.g., linear relationships with i.i.d. Gaussian noise, see [75], the mixed graphical model (MGM) of [91], or the two-variable case, see [63, 136]. As a basis for a comprehensive evaluation, these models are quite restrictive or vary strongly on their assumptions, e.g., the solely undirected edges of the mixed MGM model [91], and require manual implementation overhead as they are often not open accessible.

In summary, there exist numerous different data sets and models that allow for examining the performance of causal discovery methods within their specific scenarios. Apart from that, they do not allow the generation of observational data with varying complexity needed within a comprehensive accuracy examination of causal discovery. Especially

Table 4.1: Implementations for modeling continuous, mixed, or discrete data based upon (I) predefined benchmark data sets, (II) well-established parameterized models, and (III) functional or probabilistic models (*undirected; ‡ two-variable case; § discretized auxiliary variables).

Approach	Continuous	Mixed	Discrete
(I)	[109], [120]‡, [157]	[46]	[149]
(II)	[125], [150], [160]	[105], [157], [183]	[6], [8], [22], [80], [87]
(III)	[63]‡, [75]	[2]§, [75], [91]*	[134]‡

as this requires varying the model complexity, e.g., concerning the number of considered variables, the ratio of discrete nodes, or the number of possible discrete values, see (R2). Moreover, currently, no common framework provides a functional formalization that ensures interpretability see (R1). In contrast, we aim to establish a flexible yet well-defined and unified ground-truth model, allowing the generation of observational data under various evaluation perspectives.

4.3 The Mixed Additive Noise Model (MANM)

In this section, we introduce the mixed additive noise model (MANM) as a framework for generating causal structures with mixed discrete-continuous and nonlinear relationships. Therefore, according to (R1), we define the MANM in its functional form (Section 4.3.1). Further, regarding (R2), we provide some exemplary distribution models from continuous (Section 4.3.2), discrete (Section 4.3.3), and mixed discrete-continuous space (Section 4.3.4).

4.3.1 MANM for Modeling Mixed Discrete-Continuous Data

In general, we say that variables $V_i, i = 1, \ldots, N$ (see Section 4.1.1) of a probability space $(\Omega, \mathcal{A}, P_{\mathbf{V}})$ are discrete if they have a (finite) discrete domain, i.e., where $V_i : \Omega \to Z_i \subseteq \mathbb{R}$ with countable subset Z_i, or continuous if they have a continuous domain, i.e., where $V_i : \Omega \to \mathbb{R}$, such that all V_i have Lebesgue measurable domains in $\mathcal{A}$. Modeling causal relationships requires defining a structural causal model (SCM) that generates a variable $V_i \in \mathbf{V}$ according to the sets of possible discrete or continuous parents of V_i in $\mathcal{G}$, denoted by $\mathcal{P}^{dis}(V_i)$ and $\mathcal{P}^{con}(V_i)$, respectively.

We define the mixed additive noise model (MANM) with mutually independent noise N_i where $N_i \perp\!\!\!\perp V_j$ for all $V_j \in \mathbf{V}$ as

$$V_i = \sum_{V_j \in \mathcal{P}^{dis}(V_i)} f_{j,i}(V_j) + \sum_{V_k \in \mathcal{P}^{con}(V_i)} f_{k,i}(V_k) + N_i, \text{ for all } V_i \in \mathbf{V}, \tag{4.1}$$

with functions $f_{j,i} : Z_j \to Z_i$ and $f_{k,i} : \mathbb{R} \to Z_i$ if V_i has a discrete domain Z_i, or $f_{j,i} : Z_j \to \mathbb{R}$ and $f_{k,i} : \mathbb{R} \to \mathbb{R}$ if V_i has a continuous domain. Moreover, we require that the independent noise variable N_i either is a continuous distributed random variable, e.g., $N_i \sim \mathcal{N}(0, 1)$, or discrete distributed over Z_i with $\mathbb{P}(N_i = 0) \geq \mathbb{P}(N_i = k)$ for all $k \in Z_i$ with $k \neq 0$ if V_i is continuous or discrete, respectively.

Therefore, the proposed MANM of (4.1) extends the well-known “simple” data generating causal models and incorporates recent work on causal discovery in the two-variable model, which either are defined within a fully continuous space, e.g., see [63, 163, 194], or a fully discrete space, e.g., see [136, 134]. While we focus on causal discovery from purely observational data, the modeling of causal structures via the MANM allows for the generation of interventional data as described by [130] as well, e.g., see [57].

4.3.2 Scenario 1: MANM in the Continuous Space

In the following scenario, we provide intuitive and relatively well-established examples of the MANM assuming linear and nonlinear SCMs within the continuous domain, i.e., $\mathcal{P}^{dis}(V_i) = \emptyset$ for all $V_i \in \mathbf{V}$.

Linear Additive Noise Models: Given that $f_{k,i} : \mathbb{R} \to \mathbb{R}$ in (4.1) is linear, i.e., $f_{k,i}(x) = \beta_{k,i}x$, the MANM reduces to the most common form of SCMs [44]. In this context, the linearity assumption greatly simplifies causal discovery, e.g., see [168, 130].

In particular, when the additive noise N_i is i.i.d. standard Gaussian distributed such that (4.1) reduces to $V_i = \sum_{V_k \in \mathcal{P}^{con}(V_i)} \beta_{k,i} V_k + N_i$ with i.i.d. $N_i \sim \mathcal{N}(0,1)$ for all $V_i \in \mathbf{V}$. Then, $\mathbf{V} = \{V_1, \ldots, V_p\}$ is multivariate Gaussian distributed with mean zero and covariance matrix $\Sigma = (I_N - \mathcal{B})^{-1}(I_N - \mathcal{B})^{-1}$, where I_N is the $N \times N$ identity matrix and $\mathcal{B}$ the $N \times N$ weighted adjacency matrix with non-zero entries $\mathcal{B}_{i,j} = \beta_{j,i}$ if there is an edge $V_j \to V_i$. In this scenario, multivariate Gaussianity allows constraint-based methods to infer conditional independencies within $\mathbf{V}$ by testing for zero partial correlation, which makes causal discovery feasible for sparse graphs with up to thousands of variables, e.g., see [74]. Due to its applicability to high-dimensional settings, the linear Gaussian model has found wide use in systems biology, e.g., to infer gene regulatory networks from observational gene expression data [168]. Note that, although Gaussianity is widely assumed within linear SCMs, a non-Gaussian additive noise N_i allows for identifiability of the underlying DAG $\mathcal{G}$ beyond its Markov equivalence class, e.g., see [163].

Nonlinear Additive Noise Models: While linear models are well understood and easy to work with, causal structures within many real-world scenarios are not necessarily linear [44]. Therefore, several causal discovery methods consider nonlinear SCMs of the form $V_i = \sum_{V_k \in \mathcal{P}^{con}(V_i)} f_{k,i}(V_k) + N_i$ for all $V_i \in \mathbf{V}$, where $f_{k,i} : \mathbb{R} \to \mathbb{R}$ is not required to be linear, e.g., see [63, 178, 193]. In this context, [63] showed that, for "almost any" nonlinearities, the identifiability results of the linear non-Gaussian model can be generalized to the nonlinear case as long as the noise remains additive.

4.3.3 Scenario 2: MANM in the Discrete Space

In the discrete case, a functional relationship $f_{j,i} : Z_j \to Z_i$ provides an "interpretable" formalization of causal structures and enables the generation of observational and interventional data. Therefore, we consider that all variables $\mathbf{V}$ have a discrete domain, i.e., $\mathcal{P}^{con}(V_i) = \emptyset$ for all $V_i \in \mathbf{V}$. Then, causal relationships between discrete variables can be modeled in two different ways [136, 134]: First, V_i has the domain $Z_i = \mathbb{Z}$ with support $supp(V_i)$, such that the MANM can be defined analogously to the continuous case. Second, V_i has the domain $Z_i \subset \mathbb{Z}$, which allows to define $+$ as addition within the respective modulo ring $\mathbb{Z}/m_i\mathbb{Z}$, where $m_i = |supp(V_i)|$.

Integer Additive Noise Models: Let $V_i : \Omega \to \mathbb{Z}$, $V_i \in \mathbb{V}$, be a discrete random variable with (maybe finite) support $supp(V_i)$. In this scenario, the MANM of (4.1) reduces to $V_i = \sum_{V_j \in \mathcal{P}^{dis}(V_i)} f_{j,i}(V_j) + N_i$, for all $V_i \in \mathbf{V}$, with a function $f_{j,i} : \mathbb{Z} \to \mathbb{Z}$ and mutually independent noise N_i such that $\mathbb{P}(N_i = 0) \geq \mathbb{P}(N_i = k)$ for all $k \in Z_i$ with $k \neq 0$. Note that $f_{j,i}$ can be any probabilistic or deterministic assignment from $\mathbb{Z}$ to $\mathbb{Z}$.

For illustration, consider the two variable case $V_2 = f_{1,2}(V_1) + N_2$ with the following simplified example adapted from [136]. Let V_1 be uniformly distributed over $\{-2, -1, 0, 1, 2\}$ and let N_2 be characterized by $\mathbb{P}(N_2 = -2) = \mathbb{P}(N_2 = 2) = 0.05$, and $\mathbb{P}(N_2 = -1) = \mathbb{P}(N_2 = 0) = \mathbb{P}(N_2 = 1) = 0.3$. Then, $f_{1,2}(x)$ can either be deterministic, e.g., $f_{1,2}(x) = \lceil 0.5\ x^2 \rceil$ or probabilistic, e.g.,

$$f_{1,2}(x) = \begin{cases} Binomial(0.8, 2), & \text{if } x \in \{-2, 2\} \\ Binomial(0.5, 2), & \text{if } x \in \{-1, 1\} \\ Binomial(0.2, 2), & \text{if } x \in \{0\}, \end{cases} \tag{4.2}$$

which both induces $supp(V_1) = \{-2, \ldots, 4\}$, as implied by $f_{1,2}$ and N_2 given $supp(V_1) = \{-2, -1, 0, 1, 2\}$.

Cyclic Additive Noise Models: Following the idea of Peters et al. [136], we consider the concept of m-cyclic random variables. Therefore, let $V_i : \Omega \to Z_i = \mathbb{Z}/m_i\mathbb{Z}$, i.e., taking values in $\{0, \dots, m_i - 1\}$, such that the MANM incorporates functions $f_{j,i} : \mathbb{Z}/m_j\mathbb{Z} \to \mathbb{Z}/m_i\mathbb{Z}$. Contrary to the integer additive noise model (ANM), this scenario bounds the values each variable V_i can take to be from $\{0, \dots, m_i - 1\}$, i.e., to targeted domain $\mathbb{Z}/m_i\mathbb{Z}$.

For illustration, again consider the two-variable case $V_1 \to V_2$ with V_1 taking values $\{0, 1\}$, i.e., $V_1 : \Omega \to \mathbb{Z}/2\mathbb{Z}$, and $V_2 : \Omega \to \mathbb{Z}/3\mathbb{Z}$. Let $V_1 \sim \mathit{Bernoulli}(0.75)$ and N_2 be characterized by $\mathbb{P}(N_2 = 0) = 0.5$, $\mathbb{P}(N_2 = 1) = 0.3$ and $\mathbb{P}(N_2 = 2) = 0.2$. We then can define $V_1 \to V_2$ as $V_2 = f_{1,2}(V_1) + N_2$ with $f_{1,2} : \mathbb{Z}/2\mathbb{Z} \to \mathbb{Z}/3\mathbb{Z}$ as mapping $0 \mapsto 1$ and $1 \mapsto 2$. Moreover, the cyclic ANM enables categorical variables with discrete values that do not have any order, e.g., see [134]. For more information on the modeling of causal structures within discrete data with the ANM and further examples, see [136, 134].

4.3.4 Scenario 3: MANM in the Mixed Discrete-Continuous Space

When considering causal discovery from mixed discrete-continuous variables, mostly, a conditional linear Gaussian (CLG) model is the assumed SCM [44]. While the CLG model restricts discrete variables to have discrete parents only [2, 139], the MANM allows for both directions of causal relationships between discrete and continuous variables, e.g., following the augmented conditional linear Gaussian (ACLG) model.

Conditional Linear Gaussian Models: First, we examine how the MANM enables to generate $\mathbf{V}$ being conditional linear Gaussian (CLG) distributed. Then, the MANM of (4.1) is given by

$$V_i = \begin{cases} \sum\limits_{V_j \in \mathcal{P}^{dis}(V_i)} f_{j,i}(V_j) + \sum\limits_{V_k \in \mathcal{P}^{con}(V_i)} \beta_{k,i} V_k + \mathcal{N}(\mu_i, \sigma_i), & \text{for continuous } V_i \in \mathbf{V} \\ \sum\limits_{V_j \in \mathcal{P}^{dis}(V_i)} f_{j,i}(V_j) + N_i, & \text{for discrete } V_i \in \mathbf{V}, \end{cases} \tag{4.3}$$

where $f_{j,i}$ can be any functional assignment $f_{j,i} : \mathbb{Z} \to Z_i \subseteq \mathbb{Z} \subset \mathbb{R}$, continuous Gaussian noise $\mathcal{N}(\mu_i, \sigma_i)$, and the discrete noise term N_i as defined in Section 4.3.3. As all continuous variables are multivariate Gaussian by definition, all continuous V_i are CLG distributed given the vectors of realizations $\boldsymbol{v}^{dis}$ and $\boldsymbol{v}^{con}$ of the respective discrete and continuous parents $\mathcal{P}^{dis}(V_i)$ and $\mathcal{P}^{con}(V_i)$, i.e., we have $\mathbb{P}\big(V_i \mid \boldsymbol{v}^{dis}, \boldsymbol{v}^{con}\big) \sim \mathcal{N}\big(\sum_{v_j \in \boldsymbol{v}^{dis}} f_{j,i}(v_j) + \sum_{v_k \in \boldsymbol{v}^{con}} \beta_{k,i} v_k + \mu_i, \sigma_i\big)$, e.g., see [93]. Note that we restrict the CLG distribution to cases where neither the regression coefficients $\beta_{k,i}$ nor the variance vary given the realization of discrete parents, see [93].

Augmented Conditional Linear Gaussian Models: To overcome the restrictions of the CLG models, [93] introduced the so-called augmented conditional linear Gaussian (ACLG) model, in which discrete variables with continuous parents are generated by using the softmax function. Then, $f_{k,i} : \mathbb{R} \to \mathbb{Z}$ assigns a probability to each realization v_i within the support $supp(V_i)$, i.e., $v_i \in Z_i$ with $\mathbb{P}(V_i = v_i) > 0$, given the realization v_k of V_k. Therefore, let $f_{k,i}$ be given by the probabilistic mapping

$$f_{k,i} := \mathbb{P}(V_i = v_i | V_k = v_k) = \frac{exp(\alpha_{k,i} + \beta_{k,i} v_k)}{\sum\limits_{s \in \{1, \dots, m_i - 1\}:\ v_s \in supp(V_i)} exp(\alpha_{k,s} + \beta_{k,s} v_k)} \tag{4.4}$$

with soft-max parameters $\alpha_{k,s}$ and $\beta_{k,s}$ defined for all $v_s \in supp(V_i)$ given the realization $V_k = v_k$. Then, the MANM of (4.1) allows for generation of data according to the ACLG model via

$$V_i = \begin{cases} \sum\limits_{V_j \in \mathcal{P}^{dis}(V_i)} f_{j,i}(V_j) + \sum\limits_{V_k \in \mathcal{P}^{con}(V_i)} \beta_{k,i} V_k + \mathcal{N}(\mu_i, \sigma_i), & \text{for continuous } V_i \in \mathbf{V} \\ \sum\limits_{V_j \in \mathcal{P}^{dis}(V_i)} f_{j,i}(V_j) + \sum\limits_{V_k \in \mathcal{P}^{con}(V_i)} f_{k,i}(V_k) + N_i, & \text{for discrete } V_i \in \mathbf{V}, \end{cases} \tag{4.5}$$

with $f_{j,i}$ as defined in the context of (4.3) and the discrete noise term N_i as defined in Section 4.3.3. In this context, the MANM is flexible enough to model causal structures within discrete-continuous mixtures by incorporating various deterministic and probabilistic mappings $f_{j,i}$ from $\mathbb{Z}$ to $\mathbb{R}$, and vice versa via $f_{k,i}$. For example, consider simple step functions for $f_{k,i} : \mathbb{R} \to \mathbb{Z}$.

In summary, the MANM provides a flexible functional framework to model causal structures with various characteristics of an edge $V_j \to V_i$, see (R1)-(R2).

4.4 `MANM-CS`: A Ground Truth Framework

To provide the research community easy access to the MANM for benchmarking methods for causal discovery, see (R4), we present our reference implementation `MANM-CS`[6] implemented in `Python`. It generates mixed and nonlinear observational data according to predefined parameters (Section 4.4.1), can generate high-dimensional scenarios (Section 4.4.2), and covers various distribution models for mixed discrete-continuous data (Section 4.4.3).

4.4.1 Implementation of Data Sampling Process

Data generation of `MANM-CS` follows the common three-step approach for sampling observational data according to a CGM:

- First, a DAG $\mathcal{G}$ is sampled;
- Second, a SCM is generated according to the structure of $\mathcal{G}$;
- Third, each observation is sampled by iterating over the nodes considering the functional relationships regarding their parents.

Currently, `MANM-CS` enables generating observational data for benchmarking causal discovery in the context of mixed discrete-continuous data according to the parameters depicted in Table 4.2 on page 50.

First, (according to Table 4.2 top), `MANM-CS` generates a DAG that incorporates `num_nodes` number of ordered nodes with edge density `edge_density`.

Second (according to Table 4.2 center), nodes are chosen to be discrete with a number of classes between `min_discrete_value_class` and `max_discrete_value_class` or continuous distributed according to `discrete_node_ratio`. If the joint distribution is conditional Gaussian, as defined in `conditional_gaussian`, the first `discrete_node_ratio` × `num_nodes` are discrete, otherwise for augmented conditional Gaussian each variable is chosen to be discrete with probability `discrete_node_ratio`. According to the MANM noise terms for discrete and continuous variables are chosen to be discrete with corresponding `discrete_signal_to_noise_ratio` or normal distributed with standard deviation `continuous_noise_std`, respectively. Moreover, functional relationships for each edge in $\mathcal{G}$ are either sampled from self-chosen `functions` within the continuous space (see Section 4.3.2), follow the cyclic additive noise model within discrete (see Section 4.3.3), or defined as a soft-max and CLG model within the mixed space, respectively (see Section 4.3.4). Furthermore, the effect of continuous parents can be scaled to increase or

[6] https://github.com/hpi-epic/manm-cs

Table 4.2: Selection of parameters and their definitions for the data generation currently implemented in MANM-CS. It covers the characterization of the DAG $\mathcal{G}$ (top) and the SCM (center) and defines the sampling process of MANM-CS (bottom).

Parameter	**Definition**
num_nodes	Number of nodes N of $\mathcal{G}$
edge_density	Edge density of $\mathcal{G}$
discrete_node_ratio	Ratio of discrete nodes compared to num_nodes
discrete_signal_to_noise_ratio	Ratio of uniform noise added to discrete nodes
min_discrete_value_class	Minimum size of discrete domains $\mathbb{Z}_i$
max_discrete_value_class	Maximum size of discrete domains $\mathbb{Z}$
continuous_noise_std	Standard deviation of continuous Gaussian noise
functions	List of sample probabilities and functions $f_{k,i}$
conditional_gaussian	Flag for CGM or ACGM
beta_lower_limit	Lower limit for influence of continuous parents
beta_upper_limit	Upper limit for influence of continuous parents
scale_parents	Scaling to avoid varsortability, see [143]
num_samples	Number of observational samples n
num_processes	Number of processes used for sampling
variables_scaling	Scaling of continuous variables, e.g., normalization

decrease noise and parent's influence can be scaled to avoid varsortibility [143]. Hence, MANM-CS returns a fully parameterized CGM following the specifications of the MANM.

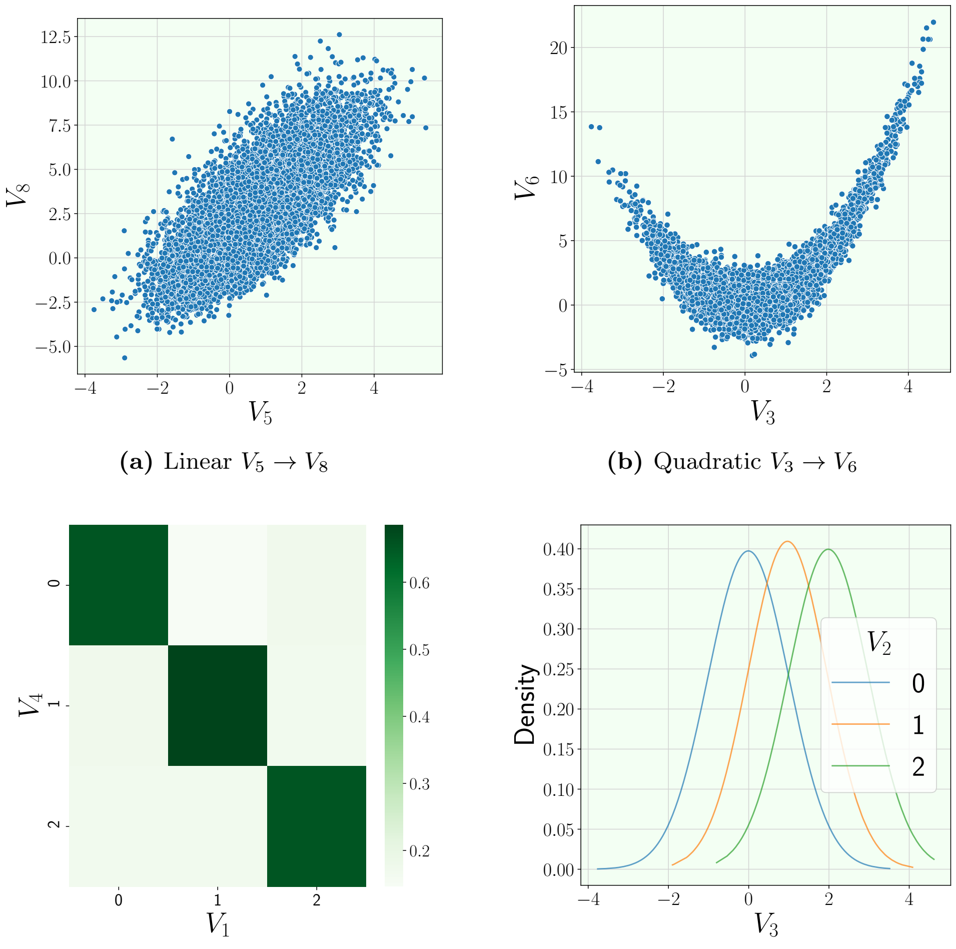
Finally (acc. Table 4.2 bottom), MANM-CS samples num_samples observations by iterating over the nodes considering the noise terms and functional relationships regarding their parents. In this context, the sampling process can be parallelized via num_processesand continuous variables of the generated observational data can be transformed according to variables_scaling, e.g., returning normalized or standardized samples.

4.4.2 Runtime Performance of Data Generation Process

To speed up data generation enabling high-dimensional scenarios, MANM-CS' data sampling can be executed in parallel by specifying the number of processes num_processes. Although ideal speed-up is not achieved, parallelization reduces data generation significantly, particularly for many nodes. For example, the execution time of one million samples generated according to a DAG with 1 000 nodes and edge density 0.4 decreases from 4 322 seconds (16 cores) over 2 214 seconds (32 cores) to 1 367 seconds (64 cores). Note, execution times are measured on a modern bare-metal server with four 32 core CPUs, with overall 2 TB RAM, varying the number of parallel processes.

4.4.3 Exemplary Characteristics of Data Generated by MANM-CS

As previously introduced, MANM-CS provides a ground-truth framework that enables examining the accuracy of causal discovery according to various distribution models. The following two examples illustrate the range of causal relationships and respective data characteristics.



(a) Linear $V_5 \to V_8$

(b) Quadratic $V_3 \to V_6$

(c) Discrete $V_1 \to V_4$

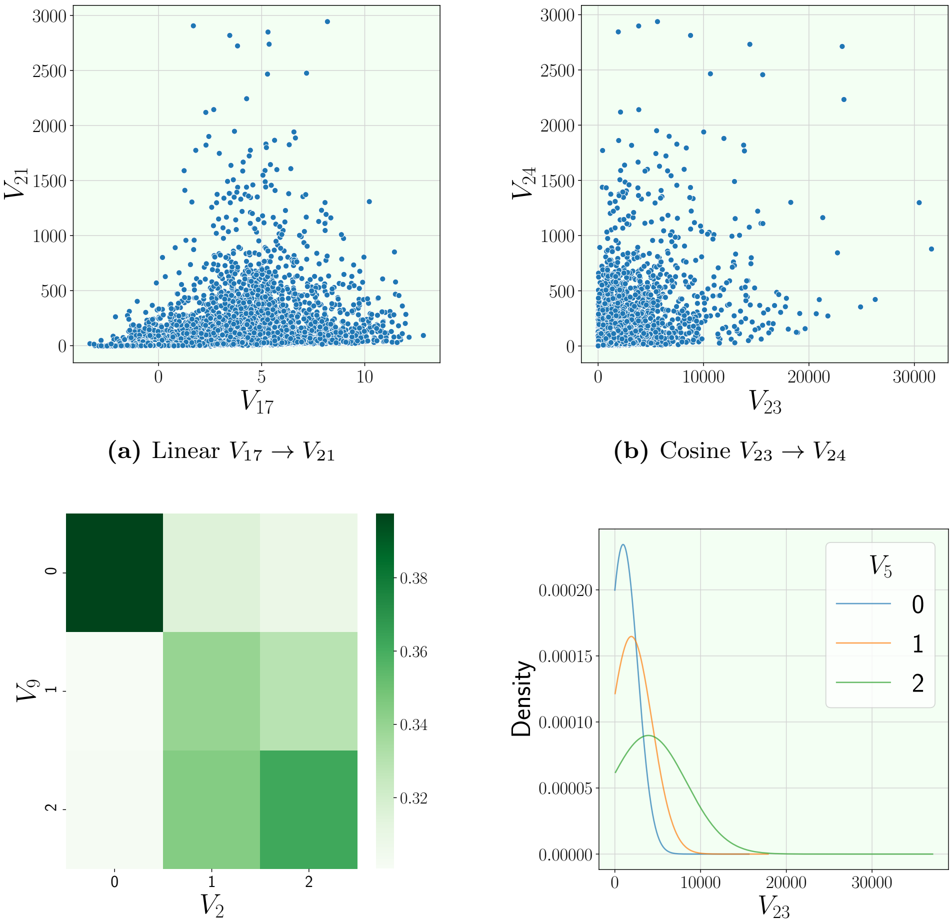
(d) Mixed $V_2 \to V_3$

Fig. 4.1: Unconfounded data distributions: (a) scatter plots for the linear edge $V_5 \to V_8$, (b) the nonlinear edge $V_3 \to V_6$, (c) a heatmap of conditional probabilities for the discrete edge $V_1 \to V_4$, and (d) a density plot of V_3 given the realization of a discrete parent V_2 for a mixed edge $V_2 \to V_3$.

Example 1: *First, we consider a small and sparse DAG $\mathcal{G}$ such that the distributional characteristics of direct causal relationships are primarily induced through the corresponding functional mappings of the underlying MANM.*

In this sense, we consider a mixed CGM (setting `conditional_gaussian`=1 *and* `discrete_node_ratio` =0.5*) of* `num_nodes` =10 *with* `edge_density` =0.4 *that includes discrete variables (with* `min_discrete_value_class` =3 *and* `max_discrete_value_class` =4*) and continuous variables with nonlinear causal relationships (defined by* `functions` ={[0.4, `linear`], [0.3, `quadratic`], [0.3, `cosine`]}*) and corresponding noise terms (*`discrete_node_ratio` =0.5 *and* `continuous_noise_std` =1.0*).*

On this basis, the data characteristics of `num_samples`=10 000 *depicted in Fig. 4.1 follow the expected evidently linear, quadratic, discrete, and mixed causal relationships of direct edges within the sparse CGM.*



(a) Linear $V_{17} \to V_{21}$

(b) Cosine $V_{23} \to V_{24}$

(c) Discrete $V_2 \to V_9$

(d) Mixed $V_5 \to V_{23}$

Fig. 4.2: Confounded data distributions: (a) scatter plots for the linear edge $V_{17} \to V_{21}$, (b) the nonlinear edge $V_{23} \to V_{24}$, (c) a heatmap of conditional probabilities for the discrete edge $V_2 \to V_9$, and (d) a density plot of V_{23} given the realization of a discrete parent V_5 for a mixed edge $V_5 \to V_{23}$.

Example 2: *Next, we consider a larger and denser DAG $\mathcal{G}$ such that the distributional characteristics of direct causal relationships may be distorted through common confounding variables.*

Accordingly, we adapt ***Example 1*** *by increasing* `num_nodes` *from 10 to 25 and* `edge_density` *from 0.4 to 0.8 while retaining all other parameters.*

On this basis, the data distributions of direct causal relationships depicted in Fig. 4.2 are now characterized by interferences of respective direct linear, cosine, discrete, or mixed causal relationships with indirect causal relationships induced through confounders within a denser CGM.

Hence, these examples not only illustrate the achieved interpretability of a causal relationship based upon a well-defined SCM, see (R1), but also demonstrate the achievable complexity of data characteristics provided by `MANM-CS`. Note, variations of the noise parameters `discrete_signal_to_noise_ratio` and `continuous_noise_std` yield further statistical dispersion. Moreover, the contrary interference on the data distribution between nodes with no direct edge in the DAG $\mathcal{G}$ may induce a visible but not existing dependence, e.g., according to confounders (see Fig. 1.2 on page 2) or chains (see Fig. 1.4 on page 6).

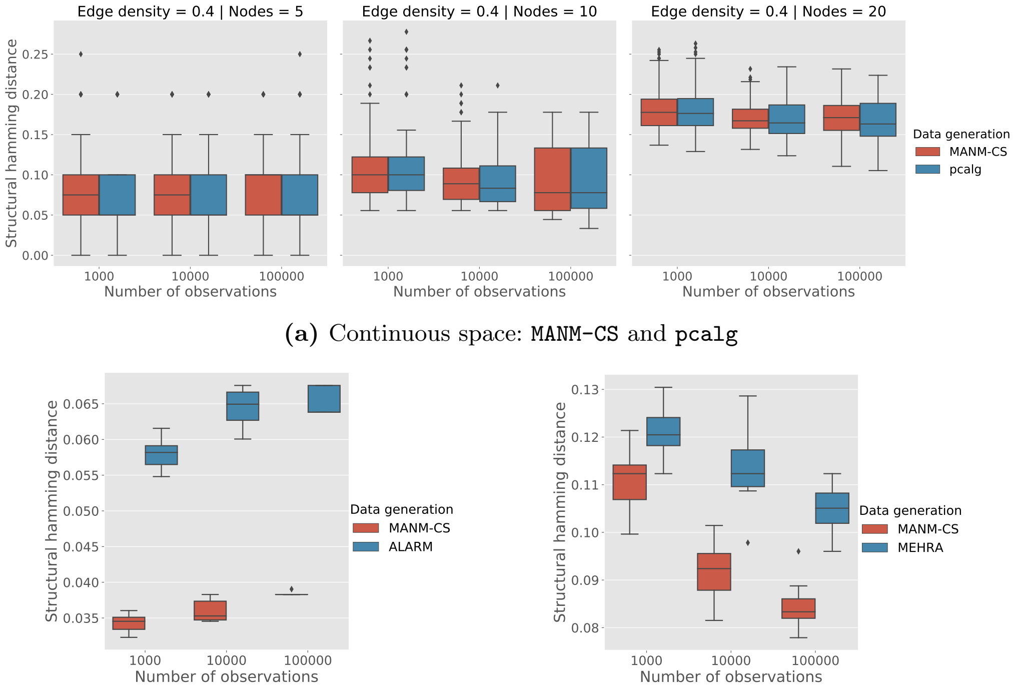
4.5 Benchmarking Scenarios and Experimental Evaluation

In this section, we demonstrate that `MANM-CS` not only covers common benchmarking approaches (Section 4.5.1), but allows for a more comprehensive evaluation of causal discovery from mixed discrete-continuous, too (Section 4.5.2).

4.5.1 Experiment 1: Comparison to other Benchmark Approaches

In this experiment, we compare large sample properties of the well-known PC algorithm [168] with appropriate conditional independence (CI) tests on observational data generated by `MANM-CS` that aims to mimic data sampled from common type (II) and (III) approaches (see Section 4.2 on page 44).

As depicted in Fig. 4.3, we show a coincidence in improvements of the structural Hamming distance (SHD) regarding the learned CPDAGs from data with increasing



(a) Continuous space: `MANM-CS` and `pcalg`

(b) Discrete space: `MANM-CS` and `ALARM`

(c) Mixed space: `MANM-CS` and `MEHRA`

Fig. 4.3: Median SHD of learned CGMs with the PC algorithm from (a) observational data sampled by `MANM-CS` or `pcalg` for the continuous space with a varying number of nodes, from (b) the Bayesian benchmark networks `ALARM` for the discrete, or from (c) `MEHRA` for the mixed space.

sample size in the context of different distribution models as introduced in Section 4.3 on page 46.

In the continuous space, see Fig. 4.3 (a), the SCM of the linear ANM (see Section 4.3.2 on page 46) allows for easy data generation following approach (III), e.g., using `pcalg` [75]. Given that the MANM is a more general model class, the SHD of CGMs learned by the PC algorithm with Fisher's z-test shows a direct coincidence on multivariate Gaussian data sampled by `MANM-CS` and `pcalg` for variations in the number of variables as well as in the number of observations.

In the discrete and mixed space, see Fig. 4.3 (b) and Fig. 4.3 (c), benchmarking of causal discovery methods is often restricted to data generated by "static" type (II) approaches, e.g., parameterized models found in the `bnlearn` repository [157]. For comparison, we generate observational data using `MANM-CS`'s capabilities for parameter adjustment to mimic the characteristics of the well-known `ALARM` [6] and `MEHRA` [183] networks. In this context, the descriptive functional restriction on modeling a causal relationship between two discrete variables within `MANM-CS` becomes recognizable. For example, `ALARM` incorporates probabilistic mappings $f_{k,i}$ between two discrete nodes V_i, V_j while `MANM-CS`'s implementation is currently restricted to the mapping of the cyclic ANM (see Section 4.3.3 on page 47). Hence, the induced independence characteristics of the variables' joint distribution are empirically stronger within data sampled by `MANM-CS`, which yield lower SHDs in comparison to the parameterized discrete `ALARM` and CLG `MEHRA` networks, respectively, see Fig. 4.3 (b) and (c). Nevertheless, the coincidence in a decreasing SHD of the CGMs learned by the PC algorithm with appropriate Pearson's X^2 (discrete, see[75]) and asymptotic mutual information χ^2 test (CLG, see [157]) for an increasing number of samples is visible for all approaches and distribution models.

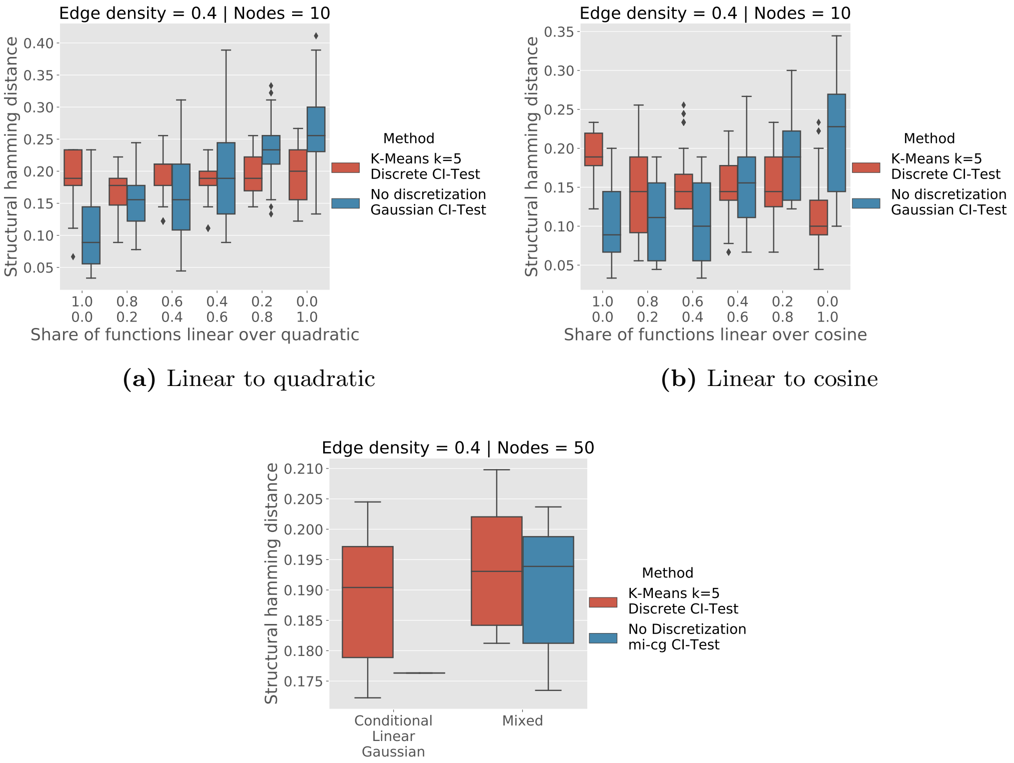
Therefore, `MANM-CS`'s capabilities not only provide enough opportunities to mimic common benchmarking approaches but also allow for more comprehensive evaluations with varying model complexity that exceed these "static" type (II) approaches.

4.5.2 Experiment 2: Benchmarking Causal Discovery with `MANM-CS`

In this experiment, we demonstrate `MANM-CS`'s capabilities for benchmarking causal discovery methods in the context of various scenarios regarding mixed discrete-continuous data. In particular, we examine the decreasing accuracy of CLG model-based causal discovery when assumptions on the causal relationships, e.g., linearity, are violated. Following [139], we include a discretization-based approach as a nonparametric baseline but consider distribution models beyond simple CLG or Lee Hastie data. Therefore, we compare the median SHD (10 runs) of learned CPDAGs with the PC algorithm [75, 157], with asymptotic mutual information χ^2 test assuming a CLG model [157], compared to Pearson's X^2 test [75], where continuous variables are discretized through the k-means algorithm [103] with $k = 5$.

Violating Linearity: We consider a violation of linearity within the CLG and its implication on the accuracy of learned CPDAGs.

In this sense, Fig. 4.4 (a) and (b) depict the median SHD of the parametric against the discretization-based approach for an increasing ratio of underlying quadratic and cosine functional relationships, respectively (`num_nodes` =10, `edge_density` =0.4, `discrete_node_ratio` =0, `num_samples`=10 000). As the asymptotic mutual information χ^2 test is based upon the partial correlation within continuous space, its good accuracy within the purely linear case decreases steadily for an increasing ratio on nonlinear functional relationships. In contrast, the discretization-based approach's accuracy behaves rather invariant in the context variations in terms of nonlinearity for both cases (a) and (b). Further, although not true generally, see [106, 139], the accuracy of the



(a) Linear to quadratic

(b) Linear to cosine

(c) CLG to mixed

Fig. 4.4: Median SHD (10 runs) of learned CGMs with the PC algorithm with asymptotic mutual information χ^2 test assuming a CLG model compared to Pearson's X^2 test where continuous variables are discretized (k-means with $k = 5$) given violation of the CLG model assumption for an increasing ratio of (a) quadratic and (b) cosine functional relationships, respectively, and (c) for considering the augmented CLG model.

discretization-based approach even exceeds the parametric CLG-based approach in the presented edge cases of mainly quadratic or cosine functional relationships.

Violating Conditional Gaussianity We consider the implication of changing from the CLG to another mixed model (see Section 4.3.4 on page 48).

In this sense, Fig. 4.4 (c) depicts the median SHD of the parametric CLG-based against the discretization-based approach under the assumption of an CLG and an augmented CLG model (`num_nodes` =50, `edge_density` =0.4, `discrete_node_ratio` =0.5, `num_samples`=10 000). Although the PC algorithm with an appropriate asymptotic mutual information χ^2 shows a significantly better accuracy within the CLG model, the accuracy is slightly exceeded by the discretization-based approach if discrete nodes are allowed to have continuous parents, i.e., within the augmented CLG model.

The above examples demonstrate the importance of validating causal discovery methods assumptions in practice and the demand for understanding the method's accuracy within specific edge cases, e.g., when assumptions on the causal relationships are violated, see [88].

4.6 Discussion

We close this chapter with a summary of our contributions (Section 4.6.1), an examination of limitations (Section 4.6.2), and future work (Section 4.6.3).

4.6.1 Summary

In this chapter, we introduced the mixed additive noise model (MANM) to provide a framework for generating causal structures within mixed discrete-continuous and nonlinear data. Its functional formalization defined in (4.1) provides an interpretable characterization of causal structures as demonstrated from a theoretical and empirical perspective (see Sections 4.3.2 to 4.3.4 and Section 4.5, respectively), see (R1). In particular, it connects well-established work of causal discovery within different distribution models, such as CLG, and allows for the generation of continuous, discrete, and mixed discrete-continuous observational data, see (R2). Due to the functional form, the MANM is flexible enough to support further extensions, e.g., to consider the generation of interventional data similar to [57], see (R3). Moreover, it allows examining methods' accuracy in case of a misspecified choice of hyperparameters or given invalidated assumptions, e.g., using an incomplete selection of $\mathbf{V}$ to model the causally insufficient case of latent variables. To provide easy access to the research community, we present our reference implementation `MANM-CS` and benchmarking scenarios, see (R4). In particular, `MANM-CS` capabilities not only provide enough opportunities to mimic common benchmarking approaches but also allow for more comprehensive evaluations with varying model complexity that exceed "static" type (II) approaches (see Section 4.5). Further, `MANM-CS` can be easily integrated into pipelines for causal discovery such as `MPCSL` [68], which allows researchers and practitioners to easily evaluate their methods.

4.6.2 Limitations

While the restriction on the SCM allows for a formalization of various causal relationships, the functional constraints induce limitations that are worth to be noticed. For example, in contrast to the assumptions of the MANM within the discrete space, see Section 4.3.3, there may not always be a functional representation of a causal relationship between discrete variables in practice [134], see Section 4.5.1. Moreover, the embedding of discrete variables into the continuous space as defined in (4.1) restricts the functional relationship to be location-related, which may be violated within real-world scenarios. In this context, possible generalizations are weakened additivity concerning the independent noise or a post nonlinearity [63, 194]. Note that if the characteristics of the data generating mechanism do not follow the MANM, the requirements of methods for causal discovery should be general enough to reveal the data generating processes approximately [44].

4.6.3 Future Work

As the MANM provides a ground truth model for generating observational data following various distribution models with nonlinear and mixed discrete-continuous data, we work on a more comprehensive empirical evaluation of several popular causal discovery methods for future work. Moreover, we aim to provide more parameters, e.g., to allow for missing values or interventional data, different noise distributions, and functional classes to enable a more fine-grained evaluation of causal discovery methods. Further, we invite the research community to participate in the extension of `MANM-CS` actively.

5

Application Scenario in Discrete Manufacturing

In the previous chapters, we introduced `MPCSL` and `MANM-CS` as tools to support the evaluation and application of causal discovery in practice, see Chapter 3 on page 29 and Chapter 4 on page 43, respectively. In this chapter, we showcase how these tools can be applied in a real-world discrete manufacturing scenario to understand unforeseen production stops. Therefore, start by motivating the application scenario and describe our contributions in more detail (Section 5.1). Further, we consider related work causal discovery in the manufacturing domain (Section 5.2). We introduce the real-world scenario of unforeseen production downtimes and the corresponding machine log data, which serves as the basis for causal discovery (Section 5.3). Moreover, we explain the transformation rules applied to the machine log data and our proposed algorithmic approach for causal discovery from the extracted data. We close this chapter with a conclusion and a discussion on limitations and future work (Section 5.5).

Contribution: Parts of this chapter have previously been published in the journal paper [70], which was noted as an equal contribution of Christopher Hagedorn and the thesis author. Christopher Hagedorn is the first author and prepared the original draft, which was developed in a cooperation project led by the thesis author. Christopher Hagedorn and the thesis author established the cooperation with the industry partner and carried out the necessary implementations as joint work. The thesis author worked out the application concepts of causal discovery, ensured the correctness of the applied mathematical concepts, particularly regarding the transformation rules and implemented algorithms, and improved the paper's material and its presentation.

5.1 Background: Machinery Production Downtimes

In this section, we motivate causal discovery from log data of discrete manufacturing (Section 5.1.1) and describe existing challenges that impede its application (Section 5.1.2). On this basis, we contribute transformation rules and a variable selection step to receive well-defined observational data, enrich causal discovery with domain knowledge to improve its accuracy, and showcase the whole causal reasoning process (Section 5.1.3).

5.1.1 Motivation for Causal Discovery from Log Data

Production downtime is one of the most significant contributors to production inefficiency [99], resulting in lost profit. While planned production downtime occurs, e.g., for scheduled maintenance based on regular schedules or predictive models [166, 77, 101, 10], unforeseen production stops are a result of failures in the production process, e.g., misconfiguration of a machine, intervention of a worker, or defective raw material. In this case, direct action from production workers is required to resolve the issue promptly and limit the financial loss [118]. Therefore, knowing the reason for the production stop and understanding the root cause supports resolving the issue effectively. Furthermore, the corresponding knowledge about the causal structures supports the machine operator to take useful precautionary measures to avoid future production stops.

Modern discrete manufacturing companies aim for increased product quality, diversified products, reduced cost, and lower manufacturing time while at the same time being faced with shortened product life cycles and global competition [97, 15, 187]. These goals are reflected in production machines that provide hundreds of configuration parameters. The introduced complexity in the machine's operation becomes challenging for the human operator to handle. The complexity rises further as, driven by the Internet of Things (IoT), an increasing amount of data is generated during production [108, 138], e.g., coming from shop floor systems, production machinery, robots, or sensors. This information is commonly stored while monitoring the production process and the product quality to detect defects. Harnessing this log data vault beyond monitoring opens the opportunity for a data-driven examination of predictive maintenance or automatic root cause analysis (RCA), e.g., for increasing production efficiency, reducing defects, or decreasing unforeseen production downtime [175, 166, 27, 188, 185, 50, 123, 144, 96]. In this context, random forests are used to derive models for predicting general machine breakdown [185, 50] or to select possible causes of faults within a manufacturing process [18]. Other approaches, based on neural networks [123, 33] or deep belief networks [96], focus on predicting specific mechanical issues within machines as an indicator for required maintenance. Clustering techniques are a way to diagnose faults in mechanical systems to retain productivity [144] or to find fault-generating combinations of machines [146].

Usually, most methods used for predictive maintenance and automatic RCA rely on associational patterns within the observational data of the production process [35]. In recent years, the emergence of methods for causal reasoning enables a data-driven examination of causal structures and causal effects beyond associational patterns within observational data [168, 130, 135, 59]. In this context, a causal graphical model depicts the respective causal structures and is the basis for causal effect estimation. Understanding the causal structures in complex manufacturing settings supports finding root causes of faults, which in practice allows machine operators to address unforeseen production stops effectively, e.g., see [84], [108], [189], or [70]. Moreover, the causal effect estimation based on learned causal structures points the machine operator to relevant adjustments or repairs within the production process, e.g., see [175] or [34]. For a detailed overview of recent advances and methods used for automatic RCA, we refer to Oliveira et al. [35]. In their work, Oliveira et al. point out the need to consider methods that allow causal conclusions and do not rely on associational patterns within the observational data. In this context, they mention the challenges of causal discovery or causal effect estimation from log data and state a research gap that focuses on the application of these methods on RCA in manufacturing processes.

In our work, we close this research gap and apply causal discovery and causal inference on log data within a real-world scenario and data from a globally operating precision mechanical engineering company. The company produces large manufacturing machines and supports the customers who operate the machinery. We focused our running example on a single machine for simplicity, yet we discovered the same patterns based on the data from similar machines of the precision mechanical engineering company.

5.1.2 Challenges in Causal Discovery from Log Data

In the scenario at hand, we find three relevant challenges to causal discovery from log data that are common in practice [106].

Challenge I (High-Dimensionality): A machine logs its configuration parameters, internal state based on sensor readings, and error messages during production. Thus, the machinery raw data contains millions of entries with several thousand different types, resulting in high-dimensional data. High-dimensional data leads to long execution times, hindering the application of causal discovery in practice [89]. Further, it increases the potential for statistical error [104].

Challenge II (Semi-Structured Raw Data): The data is recorded at different time intervals and may be stored in a semi-structured log format. Hence, the raw machinery log data needs to be preprocessed [188] before the application of causal discovery to extract the independent and identically distributed (i.i.d) observational samples [172].

Challenge III (Mixed Discrete-Continuous Data): The machinery raw log data contains a mixture of continuous variables, such as sensor measurements, and discrete variables, such as configuration parameters or error messages (see Section 1.3.2 on page 11). While recently developed methods for causal discovery work on mixed data with continuous and discrete variables [66], they are often not considered in practice due to high computational requirements.

5.1.3 Contribution

We propose a process to learn causal structures from machine log data, addressing the previously mentioned challenges. The process consists of three steps, a **preprocessing procedure**, a **procedure to learn the causal structures**, and an optional **causal effect estimation**.

Step 1 **(Preprocessing):** Within the **preprocessing procedure**, we define a set of transformation rules as a preprocessing step to obtain independent and identically distributed (i.i.d.) observations (see *Challenge II*). We integrate additional processing steps into the causal discovery procedure to handle the mixed discrete-continuous and high-dimensional data incorporating domain expertise (see *Challenges I* and *III*).

Step 2 **(Causal Discovery):** Despite these additions, the **causal discovery procedure** follows common constraint-based methods that leverage conditional independence (CI) information for learning the causal structures. Further, we define rules for edge orientation based on process-specific and engineering-specific knowledge.

Step 3 **(Causal Inference):** On the basis of the learned causal structures, **causal effect estimation** using the *do*-operator may additionally support the production worker in the RCA of production downtimes.

Our contributions to the root cause analysis (RCA) in discrete manufacturing can be summarized as follows:

- Considering a real-world production scenario, we show how the causes of unforeseen production downtimes can be revealed using causal discovery.
- We demonstrate causal effect estimation's applicability and effectiveness in an experimental regime using the learned causal structures from raw machinery log data.
- The concepts used within the proposed process are domain-independent and general enough to apply to other manufacturing industries.

5.2 Related Work on Causal Discovery from Log Data

Several studies have considered the application of causal discovery for root cause analysis (RCA) in specific use cases in the manufacturing domain [95, 84, 108, 189, 70]. Each work focuses on improvements concerning the particular use case, input data, or application. Some work incorporates preprocessing of the raw manufacturing data and utilizes domain-specific background knowledge. For an overview comparing the existing literature, see Table 5.1. To the best of our knowledge, there is no overall approach for causal reasoning in the manufacturing domain, which starts with raw log data from production monitoring and includes the application of causal inference. For general best practices for causal discovery on real-world data, we refer to Malinsky et al. [106].

In [95], see Table 5.1, Li and Shi focus on the identification of causal structures in a rolling process. They use product quality and process data to derive causal relationships to facilitate process control. Their work proposes adaptions to the PC algorithm to incorporate domain knowledge. In particular, they suggest a feature selection to relevant variables for a given causal objective to reduce dimensionality. They include temporal order information to reduce the search space. Further, they utilize engineering knowledge to discretize continuous variables from sensors and fix causal relationships that have to exist. In contrast, our study focuses on preprocessing raw machine log data to obtain observational data first, e.g., by applying transformation rules. Next, we include additional rules derived from domain knowledge during edge orientation. Further, we aim to apply general techniques for discretization when faced with a mixture of data. Lastly, we apply causal inference based on the learned causal graphical model.

Table 5.1: Comparison of related work and use cases on RCA in the manufacturing domain enriched with the focused steps of the suggested processes, the considered data, the preprocessing procedure and which domain knowledge is integration in the application scenario.

Paper & Use Case	Input Data	Preprocessing	Domain Knowledge	Application
[95] rolling process control	product quality & process data	-	causal objective, temporal order, engineering knowledge	identify causal structures
[84] chemical stirred-tank reactor	time series of process data	-	-	root cause analysis
[108] assembly line for injectors	sensor readings	remove variables: unique keys, zero variance	temporal order	root cause analysis
[189] manufacturing systems	binary manufacturing data	-	-	identify causal structures
[70] automotive body shop assembly line	failure occurrences & quality measures of car bodies	not specified	not specified	failure prediction

In [84], Kühnert and Beyerer discuss techniques to detect causal structures for root cause analysis in process technology using the example of a simulated chemical stirred-tank reactor. The work focuses on handling the time series of the process data, which differs from our approach and underlying assumptions. Yet, with changed assumptions, a time series-based approach might yield interesting results for the machine log data of our study as well.

In [108], Marazopoulou et al. focus on root cause analysis in an assembly line for injectors. Their study is based on sensor readings obtained from the assembly line. In a data preprocessing step, variables that contain unique keys or that have zero variance in their data are removed. The resulting set of variables is assumed to be continuous, and the PC algorithm is applied to learn the causal relationships. Within this step, domain knowledge, in the form of temporal order information, is applied during edge orientation. Further, they tune the significance level α according to the effect strength, based on the assumption that weak dependencies are of no interest to the domain experts, and cluster highly correlated variables into medoids to reduce the feature space. In contrast, in our study, we have a stronger focus on data preprocessing, incorporate discretization techniques to handle mixed data during causal discovery, and utilize the learned causal relationships in the causal inference step to understanding key relationships.

In [189], Ye proposes a reverse engineering algorithm to identify the underlying causal structures in manufacturing systems. The proposed approach focuses on variables with binary data only. In an evaluation, the approach outperforms several Bayesian network learning techniques. In contrast, our work is more general and not restricted to learning causal structures from binary data.

In [70], Huegle et al. demonstrate how causal structure knowledge can extend existing monitoring tools in the automotive body shop assembly lines. They use the learned causal relationships between failure occurrences and quality measures of car bodies to support technical staff in time-critical failure situations, highlighting potential root causes and predicting future failures. The authors mention data preprocessing and domain knowledge inclusion but do not specify any concrete approaches. The learned causal model from the manufacturing machine in our work could be applied similarly. Furthermore, in our work, we elucidate data preprocessing and the application of domain knowledge during the causal discovery process.

Beyond causal discovery, several research works investigate using log data for predictive maintenance [50, 166, 185]. In this context, the preprocessing of log data follows similar steps to aggregate data within time windows and select relevant features for model creation. In general, predictive maintenance aims to avoid unexpected equipment failure to reduce downtime. In contrast to these predictions, the learned causal structures support understanding the root cause and allow for a causal effect estimation in an experimental setting applying the *do*-operator.

5.3 A Real-World Production Stop Scenario

In this section, we describe a real-world scenario of production stops of a machinery production process that produce different products with a precision in the micrometer range twenty-four hours a day, seven days a week (Section 5.3.1). Further, we explain the raw data that is logged while monitoring the operation of the machinery (Section 5.3.2). In this context, we relate the three introduced challenges in causal discovery from log data to the production process and machinery data. Further, we mention assumptions and relaxations relevant to our model concerning the production scenario (Section 5.3.3).

We want to note that specific components are not described in detail for confidential reasons. However, these details are not necessary for the overall understanding of the scenario. Instead, the more general description explicitly leaves the door open for other similar applications.

5.3.1 Production Process

The machinery considered in this scenario manufactures different products operating on a two-step production process.

- Upon receiving a new manufacturing task, the machine starts manufacturing the products, first in a ramp-up phase. The machine is calibrated during the *ramp-up phase*, in which manufactured products are discarded due to low quality.
- Once the quality of manufactured products exceeds a threshold, the machine switches into the manufacturing phase. The specified number of products, as defined in the current manufacturing task, is manufactured during the *manufacturing phase.*

After finishing the current manufacturing task, the machine continues with the next manufacturing task. The entire process is monitored, e.g., to determine the quality of the products to switch from the ramp-up to the manufacturing phase, and log messages are obtained and stored. The production process is interrupted when unforeseen production stops occur.

Manufacturing Tasks: We denote the set of all manufacturing tasks with T. Hence, a new *manufacturing task* $t \in T$ is issued for each product change with a set of specified configuration parameters. As previously mentioned, a manufacturing task is split into two separate phases:

1. First, the task enters its *ramp-up phase* t^c to calibrate the configuration parameters and assure the quality of the product. During this phase, both the machine operator and the machine itself optimize the machine settings to adjust for an accurate manufacturing result.
2. Second, upon successful execution of the ramp-up phase, the task enters its *manufacturing phase* t^e, which executes the defined task and manufactures the predefined number of corresponding products from the provided input material.

Accordingly, the set of tasks T can be split into subsets for the ramp-up and manufacturing phases, denoted with t^c and t^e, respectively.

Log Messages: During operation, internal machine parameters are constantly monitored within different machine subsystems during the ramp-up and manufacturing phases.

During monitoring, messages are logged and stored in a single event log for all machine parameters. In this context, logging occurs event-based (*Challenge II*), and we denote the set of all messages with M. Hence, measurements provided through sensors are continuously logged at a sampling frequency of multiple values per second. Furthermore, event-based logging occurs either in the case that a predefined threshold is violated or in the case that a machine operator interacts with the machine. The majority of the messages are interactions of the machine operator with the machine, or non-critical issues, e.g., “low oil level”.

Moreover, there exist messages which indicate changes in a task's phase. In particular, we distinguish between messages for the beginning of the ramp-up phase, the beginning of the manufacturing phase, and the end of the manufacturing phase. Hence, we define the following disjoint subsets of the messages M. The set of messages $M^{bc} \subset M$ contains all messages indicating the *start of the ramp-up phases*, the set of messages $M^{be} \subset M$ contains all messages indicating the *start of the manufacturing phases*, which also represents the ends of the ramp-up phases. Further, the set of messages $M^{ee} \subset M$ contains all messages indicating the *end of the manufacturing phases.* All remaining messages are contained in $M^k \subseteq M$.

Production Stops: A threshold violation with a respective message directly results in an unforeseen production stop for several parameters. While this informs a machine operator about a production stop and its direct trigger, it does not inform the machine operator of any causes that lead to the threshold violation.

5.3.2 Description of Raw Machinery Log Data

The raw machinery data logged for monitoring is semi-structured, i.e., individual log data entries, called log messages, vary in their structure.

In the following, we define a joint model for all log messages as shown in the class description for a `LogMessage` in Fig. 5.1, which specifies mandatory and optional fields.

The two mandatory fields, `time`, and `message_id` occur in all log messages and are inevitable for the transformation to observational data. The `time` field contains a timestamp of the message's occurrence, and the `message_id` contains a unique identifier of the message type. Together both fields uniquely identify each log message instance.

Optionally, a log message object contains the three fields `location`, `parameter_value`, and `message_description`. The `location` contains machine-specific location information, for example, describing the position within the machine according to components. The `parameter_value` contains a value from a continuous or discrete domain (*Challenge III*), which is sent for certain message types such as sensor readings. The `message_description` provides additional detail for a message. Note that both `parameter_value` and `message_description` have varying formats, depending on the machine's component and implementation.

In our example, we focus on a single machine, which logged over 40 million entries within one year. Based on the set of logged messages M, we reconstruct 25 729 tasks $t \in T$ and determine over 6 000 unforeseen production stops. Further, the logged messages contain 2 330 distinct message types, i.e., unique `message_ids`. In the context of a causal graph $\mathcal{G}$ (see Section 1.2 on page 3), we will map a unique `message_id` to a variable $V_i \in \mathbf{V}$. Consequently, the large number of variables constitutes *Challenge I*.

Message types can be classified into four distinct classes based on domain knowledge. This classification supports the transformation to observational data, particularly for many message types, as handling them manually becomes too time-consuming. The four classes are task *configuration parameters* C, *non-critical operational messages* O, *production stop messages* S and *continuous measurements* Q.

```
class LogMessage:
    def __init__(self, time, sub_id, msg_id,
                 location=None, param=None, msg_desc=None):
        self.time                = time
        self.message_id          = msg_id
        self.location            = loc
        self.parameter_value     = param
        self.message_description = msg_desc
```

Fig. 5.1: Representation of a log message object `LogMessage` with its defined mandatory and optional fields. Optional fields of `LogMessage` are marked with `=None` in the `__init__` function.

- The *configuration parameters* C subsume all message types that refer to product specifications defined for a manufacturing task.
- The class of *non-critical operational messages* O contains all message types that signal standard production process flow, such as the message types for changes in the task's phases or message types indicating rotation of the product.
- The class of *production stop messages* S contains all message types that signal an unplanned production stop during the production process, e.g., due to issues with amounts of material picked up by subsystems of the machine.
- Lastly, the *continuous measurements* Q subsume all message types that relate to continuously obtained measurements, such as sensor readings.

We utilize this classification schema throughout the paper, e.g., when naming distinct messages, respectively variables. Note that the messages for changes in the task's phase are classified as "non-critical operational messages".

5.3.3 Assumptions and Relaxations of Causal Discovery

For a gentle introduction to causal discovery and its assumptions, see Section 1.2 (page 3) and, for an elaborate background on causal discovery, we refer to [168].

According to the theoretical background on causal discovery, we assume that the true underlying directed acyclic graph (DAG) $\mathcal{G}$ is *acyclic*, i.e., we limit the production stop scenario to exclude feedback cycles such that messages cannot mutually influence each other. Further, we assume *causal sufficiency*, i.e., that the set of messages contains all relevant information for the production process such that we may rule out unmeasured confounding variables. Thus, we also cannot account for outside influences such as fluctuations of the temperature in the factory, see Section 1.2 (page 3). Furthermore, we assume *faithfulness* and that the *causal Markov condition (CMC)* hold, i.e., (in)dependencies between machinery measurements, configuration parameters, and messages arise not from incredible coincidence but rather from the structure of the DAG $\mathcal{G}$, and vice versa.

Concerning the production process itself, we assume that the underlying mechanisms logging the data do not change within the available time period, e.g., due to software updates or hardware changes. Note that this assumption is required to receive independent and identically distributed (i.i.d.) observational samples necessary for causal discovery. Hence, we do not cover drift in the underlying model, respectively the DAG $\mathcal{G}$. Similarly, when generalizing the results to other machines of the precision mechanical engineering company, we assume that the same operations and processes occur. Otherwise, the causal structures have to be learned for each machine individually.

5.4 From Raw Machinery Log Data to Causal Insights

In this section, we detail our fairly general process for causal discovery from raw machinery log data, as shown in Fig. 5.2, and apply methods of causal inference based upon the learned causal structures. Note, while we will explain and apply this process in the context of our scenario described in Section 5.3, we would like to highlight that the process' components and concepts to be used are of general type, i.e., they are not limited to our scenario and can also be adapted to analyze the log data of other applications.

In particular, the information provided by the domain expert marked with DK in Fig. 5.2 requires adaption to correspond to the considered application. While some domain knowledge is required, see bold elements in DK data structures in Fig. 5.2, other domain-specific information is optional. If optional domain knowledge is not provided, the process resorts to defaults, affecting the quality and interpretability of results.

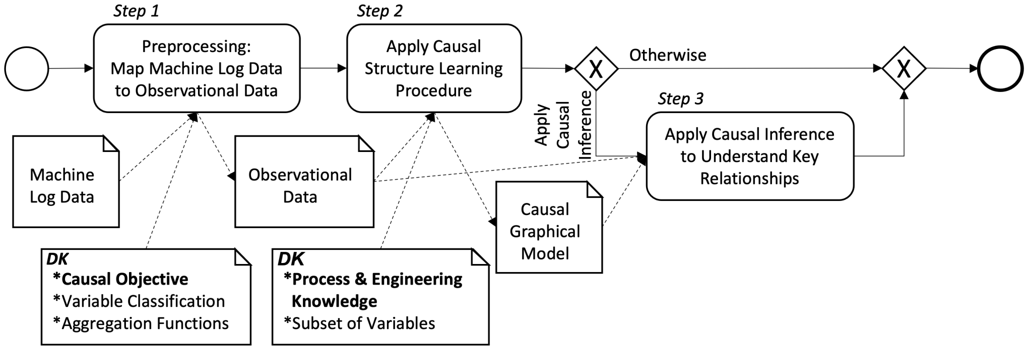


Fig. 5.2: Outline of the process for causal discovery from raw machinery log data. Rectangles with rounded corners depict process steps, while rectangles with a folded corner depict data structures. Data structures marked with *DK* represent a list of information provided by a domain expert. Bold list elements are required; other list elements are optional. Providing optional information avoids defaults and improves the quality and interpretability of results.

Note that the following steps have been developed with the tool support provided by `MPCSL` and `MANM-CS`, which helped improving the accuracy and applicability of the proposed procedure. Hence, this case study not only demonstrates the opportunities of causal discovery in practice but also emphasizes the contribution of **C1** to answer **RQ1** (see Section 1.4.1 on page 15).

In ***Step 1 (Preprocessing)***, see Section 5.4.1 on page 65, we preprocess logged raw machinery data, mapping the log messages to individual observations, to obtain observational data, addressing *Challenge II.* This step requires the domain experts' input of a causal objective, which guides the mapping approach. Optionally, the domain expert can classify the obtained variables from the log data and assign an aggregation function to each class of variables to improve the mapping step. The observational data is input for the second.

In ***Step 2 (Causal Discovery)***, see Section 5.4.2 on page 68, we apply algorithms to learn the underlying causal structures. Given the real-world data characteristics, we include variable selection and data discretization into the causal discovery procedure, addressing *Challenge I* and *III.* Furthermore, providing process and engineering-specific knowledge in the form of relations between classes of variables allows extending existing orientation rules to reflect domain knowledge.

In the optional ***Step 3 (Causal Inference)***, see Section 5.4.3 on page 74, we apply the *do*-operator to understand key relationships. In this context, the learned causal structures are input to methods of causal inference, which finally allow revealing causal effects in the considered production scenario.

5.4.1 *Step 1 (Preprocessing)*: Mapping of Raw Machinery Log Data to Observational Data

In the following, we provide details on our transformation procedure, i.e., ***Step 1 (Preprocessing)***, which maps machine log data to observations, tackling *Challenge II* (see Section 5.1.2 on page 58).

In our transformation procedure, we define time windows based on the context of a domain-specific causal objective. We map the machine log messages into the time windows to extract the set of relevant variables for the DAG $\mathcal{G}$ and to transform the log messages within a time window into one observation through the application of three defined transformation rules. Finally, we apply this transformation procedure in our production stop scenario to obtain well-defined observational samples, which results in 25 729 observations of 1 903 variables.

Transformation Procedure: The transformation to derive observational data from raw machinery log data requires the input of a *causal objective* that is based on the domain knowledge of the machine operator. Further, optional inputs are a *variable classification* and specific *aggregation functions*.

The transformation procedure uses the context of the causal objective to define *time windows* for individual observations. Then, all machine log messages are mapped into a time window. These mapped machine log messages are used to extract the set of N variables $\mathbf{V} = (V_1, \ldots, V_N)$ corresponding to the nodes of the DAG $\mathcal{G}$.

Finally, through the definition and application of three transformation rules, the log messages within each time window are processed together with the set of variables to determine the observations. Note, if provided, variable classification and aggregation functions can replace defaults within the transformation rules.

In the production process at hand, the objective to understand production stops is defined in the context of the set of manufacturing tasks T (see Section 5.3.1 on page 62). Each manufacturing task comprises of a ramp-up phase t^c and manufacturing phase t^e. Accordingly, the following three time windows are defined for each manufacturing task.

Definition of Time Windows: A time window δ_t is defined for a task $t \in T$. A second time window δ_t^c represents the task's ramp-up phase, while a third time window δ_t^e is defined for the task's manufacturing phase.

The calculation of the time windows is based upon the timestamp *time* of the corresponding messages drawn from the sets M^{bc}, M^{be}, M^{ee} for the task t, as shown in Eq. (5.1). The function `time()` retrieves the timestamp from the selected message. Note that, for each task t, there exists exactly one message $m_t^{bc} \in M^{bc}$, one message $m_t^{be} \in M^{be}$, and one message $m_t^{ee} \in M^{ee}$. Hence, let

$$\begin{aligned}\delta_t &:= \texttt{time(}m_t^{bc}\texttt{) until time(}m_t^{ee}\texttt{)}\\ \delta_t^c &:= \texttt{time(}m_t^{bc}\texttt{) until time(}m_t^{be}\texttt{)}\\ \delta_t^e &:= \texttt{time(}m_t^{be}\texttt{) until time(}m_t^{ee}\texttt{)}.\end{aligned} \tag{5.1}$$

Lastly, we denote the set of all time windows δ_t for all tasks T as $\varDelta$.

Mapping Messages to Time Windows: The time windows δ_t, δ_t^c or δ_t^e, see Eq. (5.1), for each task t are used to map all messages $m^k \in M^k$ to the task's ramp-up phase t^c or to the task's manufacturing phase t^e. Therefore, for each message m^k first, its timestamp *time* from the log message object is selected.

Next, while m^k is not mapped to a task t, all time windows $\delta_t \in \varDelta$ are iterated, and it is checked if *time* of m^k is within δ_t. Once the condition is met, it is specified whether the *time* is within the corresponding time windows δ_t^c or δ_t^e. Accordingly, the log message m^k is mapped to task t corresponding to δ_t and specifically to either t^c or t^e. Note that messages outside these time windows, e.g., during general maintenance, are ignored.

As a result, two lists of log message objects for each task t are returned. One list contains all messages m^k corresponding to t^c, while the other list contains all messages m^k one corresponding to t^e.

Extracting Variables from Messages: The distinct `message_id`s of all log message objects (see Fig. 5.1 on page 63) make up the potential set of variables $\mathbf{V} = (V_1, \ldots, V_N)$ corresponding to the nodes of the DAG $\mathcal{G}$. Yet, given the previous transformation steps, we limit the set of variables. Therefore, we consider variables for which a `message_id` occurred within any time windows δ_t. Further, we distinguish for each `message_id` whether it occurred within the ramp-up or manufacturing phase. Note that the same unique `message_id` is used in both phases. Therefore, we search the two lists of log message objects for each task t and extract the distinct `message_id`s annotated per phase. The resulting set of `message_id`s defines the set of variables $\mathbf{V} = (V_1, \dots, V_N)$ for each observation.

Transformation Rules: Based on the set of variables $\mathbf{V}$ for all observations, we define the following three *transformation rules* to transform the mapped log messages per time window to independent and identically distributed observations.

Hence, the three rules are applied for each task $t \in T$ and its corresponding time window $\delta_t \in \Delta$, respectively δ_t^c and δ_t^e.

- **Transformation Rule 1:** For each variable $V_i \in \mathbf{V}$, $i = 1, \dots, N$, for which there exists only a single message in M^k within the time windows δ_t^c or δ_t^e, the value of V_i is set for the corresponding task t in the following way. If the log message object contains a parameter, the parameter's value is used as a value for V_i. Otherwise, V_i is set to 1.
- **Transformation Rule 2:** For each variable $V_i \in \mathbf{V}$, $i = 1, \dots, N$, for which there exists no message in M^k within the time windows δ_t^c or δ_t^e, the value of V_i is set to be 0.
- **Transformation Rule 3:** For each variable $V_i \in \mathbf{V}$, $i = 1, \dots, N$, with multiple occurrences of a message M^k within the time windows δ_t^c or δ_t^e, an aggregation function is applied to calculate the value for V_i.

For each variable $V_i \in \mathbf{V}$, a different aggregation function may be specified. If the message M^k contains a parameter, well-known aggregation functions, such as average, minimum, maximum, or last value, can be chosen. Otherwise, if the messages M^k contain no parameter, the choice of aggregation function is limited to counting occurrences or one-hot encoding the occurrence.

The Transformation Rule 3 resorts to default functions, i.e., average if the messages contain a parameter and counting if the messages contain no parameter. Yet, the defaults are overwritten if domain experts provide the optional list of aggregation functions for variables.

In case of a high number of variables, specifying an aggregation function for each variable becomes cumbersome. Therefore, we allow for aggregation functions to be specified for classes of variables if a variable classification is provided as input by the domain expert. Note that the choice of the aggregation function depends on the causal objective and influences the interpretability of the subsequent process steps. Hence, domain expertise is invaluable to the choice of a suitable aggregation function.

Real-World Production Stop Scenario: In the production stop scenario of the machine manufacturer, the transformation procedure from machine log data to observations results in 1 903 variables with 25 729 observations, corresponding to individual manufacturing tasks $t \in T$.

The variables stem from both the ramp-up and manufacturing phases. Out of the 2,330 distinct message types 427 did not occur in any time window $\delta_t \in \Delta$. With close to 2 000 variables choosing an aggregation function for each variable is infeasible for a domain expert.

Therefore, domain experts provided aggregation functions together with a variable classification of four classes (see Section 5.3.2 on page 63). In detail, the last value is used for *configuration parameters C*. The occurrences are counted for *non-critical operational messages O*. *Production stop messages S* are always binary, indicating whether a stop

Table 5.2: Excerpt of the derived observational samples after applying the transformation procedure to raw machinery log data. Each row marks one observational sample. Each of the columns 2-12 marks one variable. Observational data from a selection of eleven variables is presented.

task id	C_x	C_y	C_z	O_1	O_2	S_1	S_2	$Q^{\{e\}}_{speed}$	$Q^{\{c\}}_{speed}$	$Q^{\{e\}}_{number}$	$Q^{\{c\}}_{number}$
1	1 020	710	11	0	0	0	1	9 833	11 000	1	101
2	890	641	11	0	0	0	0	14 000	12 000	412	3
3	915	640	19	0	1	0	0	6 825	13 000	172	101
...	...	...	...	...	...	...	...	...	...	...	...
25 729	1 020	710	14	3	0	1	0	13 000	7 800	1	404

occurred or not. Lastly, *continuous measurements* Q use the mean parameter value from all messages that occurred within the current task's phase. An exemplary excerpt is shown in Table 5.2.

5.4.2 *Step 2 (Causal Discovery)*: Application of Causal Discovery Procedure to the Observational Samples

In the following, we outline the application of the causal discovery procedure, i.e., ***Step 2 (Causal Discovery)***, to learn the underlying causal structures of the machinery production process. We introduce the procedure tailored to the manufacturing domain and highlight steps that address *Challenge I* and *III* (see Section 5.1.2 on page 58).

The procedure takes the observational data as input to output the learned causal structures. Furthermore, structured information on the process and engineering knowledge is required from the domain expert to facilitate the learning procedure. Optionally, the domain expert can specify a subset of variables to limit the size of the CGM. Finally, we report the learned causal structures of the production stop scenario in two different settings, i.e.,

- ***Case I (Application)*** on the entire variable set and
- ***Case II (Validation)*** on a subset of variables selected by a domain expert.

Causal Discovery Procedure: As depicted in Fig. 5.3, we propose a causal discovery procedure based on the well-known PC algorithm [168] (see Section 1.2 on page 3). The procedure incorporates two steps that address the manufacturing domain-relevant *Challenges I* and *III* (see Section 5.1.2 on page 58) and augments the edge orientation step of the PC algorithm using domain knowledge.

The procedure starts with the **Select Variables** step addressing *Challenge I.* This step extracts a subset of variables from the observational data specified by a domain ex-

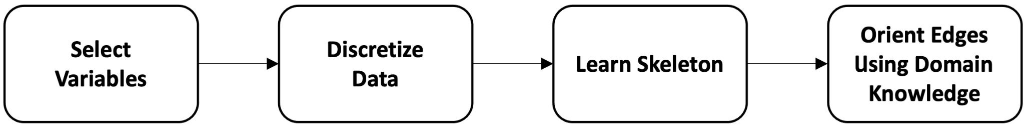


Fig. 5.3: Steps of the causal discovery procedure, ***Step 2 (Causal Discovery)***, which consists of the two steps, i.e., **Select Variables** and **Discretize Data**, that address manufacturing domain-relevant challenges and two steps of the PC algorithm, i.e., **Learn Skeleton** and **Orient Edges Using Domain Knowledge**. Note that in this procedure, the edge orientation from the PC algorithm is augmented with domain knowledge.

pert. Next, in the **Discretize Data** step, the observational data of the selected variables is discretized, addressing *Challenge III.* The skeleton $\mathcal{C}$ of the DAG $\mathcal{G}$ is estimated from the discretized observational data in the **Learn Skeleton** step. The resulting undirected edges in the skeleton $\mathcal{C}$ are oriented using common orientation rules of the PC algorithm and domain knowledge within the last step **Orient Edges Using Domain Knowledge**.

Select Variables: In the first step of the procedure, a domain expert can select a subset of variables denoted by $\mathbf{V}^S \subseteq \mathbf{V}$ that is relevant to the given causal objective.

The variable selection step has two goals.

- First, considering only a subset of variables can effectively reduce the search space within the **Learn Skeleton** step, resulting in faster execution times and poses a solution for *Challenge I.*
- Second, removing variables having similar meaning or variables known to have no relevant impact [171] reduces the effects of noise within the dataset.

The selection requires expert knowledge, as no common rule, i.e., for removing semantically dependent variables, exists [186]. Further, selecting variables requires caution to avoid violation of the *causal sufficiency* assumption. The subset of variables $\mathbf{V}^S$ is said to be *causally sufficient* if $\mathbf{V}^S$ incorporates all common causes or confounders that causally influence more than one variable in $\mathbf{V}^S$ [167] (see Section 1.2 on page 3).

Discretize Data: In the second step of the procedure, the observational data of variables $V_i \in \mathbf{V}^S$ with continuous data is discretized. Hence, as a result of this step, the observational data for all variables $V_i \in \mathbf{V}^S$ is assumed to be discrete. While discretization results in loss of information [32, 106] it allows to handle datasets with a mixture of continuous and discrete variables, addressing *Challenge III.*

In this context, we refer to the experiments using `MANM-CS` in Section 4.5 on page 53, which indicates that the discretization-based approach may exceed competitive approaches, particularly if the corresponding assumptions are violated. Moreover, this discretization is particularly relevant for higher-dimensional datasets, where the alternative to use CI tests for mixed data,e e.g., see [66, 69, 112], is impeded due to their long runtimes. Moreover, discretization simplifies causal inference as we need to only consider causal effect estimation using the *do*-operator between discrete variables.

There exist approaches that consider domain-specific discretization [97]. Yet, with a larger number of variables, such an approach becomes tedious and time-consuming for any domain expert and thus infeasible in our case. Therefore, general discretization techniques, such as clustering, e.g., k-means clustering, or binning, e.g., equal-width binning, must be considered. These approaches are applied to each variable and assign the observational data of the variable to one of k representatives. The choice of k directly impacts the amount of information loss, with smaller values of k reducing the significance of the information flow [73]. Yet, for large values of k, the degree of freedom within the CI test increases, demanding a higher number of observations or leading to poorer quality of the learned causal structures [29]. Therefore, a careful choice of the parameter k is required.

Learn Skeleton: In the third step of the procedure, the discrete observational data for the subset of variables $\mathbf{V}^S$ is used in the first phase of the PC algorithm, referred to as skeleton discovery. Through the repeated application of appropriate CI tests, which is directly determined by the underlying data distribution [28], the undirected skeleton graph $\mathcal{C}$ of the DAG $\mathcal{G}$ is learned. For the discrete observational data, this step applies Pearson's χ^2 test [132] as the CI test. Further, to handle high-dimensional datasets efficiently on modern hardware, this step utilizes an existing parallel implementation [53, 153, 158].

Orient Edges Using Domain Knowledge: In the final step of the procedure, the undirected edges of the skeleton $\mathcal{C}$ are oriented to derive the CPDAG of $\mathcal{G}$ (see Section 1.2 on page 3).

Therefore, the well-known Meek's orientation rules of the PC algorithm are applied, see [115, 74, 167], before the incorporation of domain knowledge according to an approach that was originally introduced by Meek [115], too.

In this context, we introduce the necessary standard notation for edges in a graphical model, e.g., [168, 130]. In particular, a graph $\mathcal{G}$ is defined as

$$\mathcal{G} = (\mathbf{V}, \mathbf{E}) \qquad \text{with } \mathbf{V} = (V_1, \ldots, V_N) \text{ and } \mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}, \tag{5.2}$$

where $\mathbf{V}$ and $\mathbf{E}$ denote the set of N nodes and the set of edges between nodes, respectives. As introduced in Section 1.2 on page 3, each node V_i, $i = 1, \ldots, N$ relates to an observed variable. Hence, an edge is defined as $(V_i, V_j) \in \mathbf{E}$, for $i, j = 1, \ldots, N$ and $j \neq i$. Therefore, a directed edge $V_i \rightarrow V_j$ is encoded as $(V_i, V_j) \in \mathbf{E}$ and $(V_j, V_i) \notin \mathbf{E}$, and a undirected edge $V_i - V_j$ is encoded as $(V_i, V_j) \in \mathbf{E}$ and $(V_j, V_i) \in \mathbf{E}$.

Then, domain knowledge DK, also called background knowledge, is defined as a pair $DK = (F, R)$, where $F \in \mathbf{E}$ is a set of forbidden directed edges and $R \in \mathbf{E}$ is a set of required directed edges [115]. The original algorithm by Meek [115] fails if the learned complete partially directed acyclic graph (CPDAG) based on the application of the standard orientation rules of the PC algorithm is not consistent with the DK, i.e., learned oriented edges violate edges in R or F.

We propose to relax this consistency constraint and consider DK as recommendations, i.e., we consider R as recommended edges, and further exclude any forbidden directed edges, i.e., $F = \emptyset$. We process DK, respectively R, as follows. Note that at this point, we start with the skeleton learned with the PC algorithm and oriented according to the detected v-structures (see Section 1.2 on page page 3). In the following, we denote this partially directed acyclic graph with $\mathcal{G}_v$.

Then, for each edge $E = (V_i, V_j) \in R$ check if $E \notin \mathbf{E}$ of $\mathcal{G}_v$. If the edge E does not exist in $\mathcal{G}_v$, add $E \in R$ to a list of edge violations. Otherwise, if $E = (V_i, V_j)$ exists in $\mathcal{G}_v$ and there exists $E' \in \mathbf{E}$ of $\mathcal{G}_v$ with $E' = (V_j, V_i)$, i.e., we have an undirected edge in the partially directed acyclic graph $\mathcal{G}_v$, remove E' from $\mathcal{G}_v$ and close orientations following Meek's edge orientation rules of the PC algorithm [115]. The list of violations is returned and allows for further investigation by the domain expert. Thus, the returned list of violations may introduce feedback loop into the causal structure learning procedure, which accounts for erroneous assumptions in the variable selection, i.e., due to violations of the semantic independence of variables [106], or erroneous assumptions in the data discretization.

Instead of relying on input from a domain expert for individual edges to define the set of recommended edges R, we define domain-specific rules to determine R. Generally, these rules can be based on temporal ordering, engineering details, or process knowledge [97].

For the production process in our case study, we utilize process and engineering knowledge that is passed as input to the causal structure learning procedure to define the following rules.

- ***Rule 1:*** For each variable $V_i \in \mathbf{V}$, with V_i stemming from the ramp-up phase and $V_j \in \mathbf{V} \backslash \{V_i\}$, with V_j stemming from the manufacturing phase and $i, j = 1, \ldots, N$, if there exists an edge between $V_i - V_j$ in $\mathcal{C}$, add a recommended directed edge $V_i \rightarrow V_j$ into R.
- ***Rule 2:*** For each variable $V_i \in \mathbf{V}$, with V_i stemming from a configuration parameter c and $V_j \in \mathbf{V} \backslash \{V_i\}$, with V_j stemming from either a non-critical operational message o, a production stop message s or a continuous measurement m and $i, j = 1, \ldots, N$, if there exists an edge between $V_i - V_j$ in $\mathcal{C}$, add a recommended directed edge $V_i \rightarrow V_j$ into R.

Hence, ***Rule 1*** considers process knowledge based on the context of the causal objective, recommending to orient all existing edges between variables from the ramp-up phase and variables in the production phase to orient towards the variables in the production phase. moreover, ***Rule 2*** considers engineering-specific information, stating that variables representing configuration parameters should not be influenced by any variables representing information obtained during the production process, regardless of the task's phase. Hence, the edge is oriented away from the configuration parameter. Note that there is no particular rule covering temporal ordering.

Real-World Production Stop Scenario: The following analysis has two goals. First, we want to illustrate the application and value of learning the causal structures to understand unforeseen production downtimes in manufacturing. Second, we aim to validate our proposed process to determine accurate causal structures from raw machinery log data.

Therefore, we consider two different settings in the production stop scenario. First, in ***Case I (Application)***, we take the entire set of variables $\mathbf{V}$ skipping the variable selection. Here, we demonstrate the applicability of our proposed process to derive causal structures from log data. Since validation of the process' results on the entire set of variables with over 3 million possible causal relationships is infeasible for domain experts, we consider a second case. In ***Case II (Validation)***, we let a domain expert select a validated subset $\mathbf{V}^S$ containing 11 variables. Within this subset of variables, the domain experts fully understand the mechanisms and can judge if our process correctly learned the structures, detected false positives, or is missing any relevant structures. Besides, we use the resulting causal structures from $\mathbf{V}^S$ as input for methods of causal inference to keep this illustrative example simple.

In both cases, we discretized the sets of variables (see the part on **Discretize Data**). We considered a standard equal-width binning and k-means clustering for the discretization step. We could not find a significant difference in the resulting learned causal graphs. Yet, binning showed a marginally improved result on the validated subset. Hence, for both cases, we consider the results using equal-width binning. Further, through parameter tuning, we determined a value of $k = 5$ to produce the best results on the validated subset $\mathbf{V}^S$, which is also used as a parameter for the case of learning on the entire set of variables $\mathbf{V}$. In the subsequent skeleton learning step, we set the decision threshold for each CI test α to 0.01, which is common in practice [20].

Case I (Application): In the following, we consider the learned causal structures on the complete dataset from the globally operating machine manufacturer, omitting the step to select variables based on domain knowledge. Given the larger number of variables and a missing gold standard, validation by a domain expert becomes infeasible, and we cannot report overall accuracy metrics for the entire set of variables. Yet, we report the accuracy of the domain knowledge-based edge orientation. Further, to understand the learned causal graph, we report metrics common to describe graphical models. Besides the number of variables and learned edges corresponding to the causal structures, we report each node's maximum and average in- and outdegree. The indegree is defined as the number of incoming edges, whereas the outdegree describes the number of outgoing edges. Thus, the maximum and average in- and outdegree over the entire graph allow getting an understanding of the complexity of the learned causal model. For example, a low average in- and outdegree describe sparse graphical models. The parameters characterizing the learned causal relationships on the entire dataset are shown in Table 5.3 on page 72.

The nodes in the learned graph represent the $1,903$ variables, which are connected by 550 edges, counting both directed and undirected edges. In total, $1,068$ nodes have no edges, meaning no causal relationship to any other node. The maximum indegree $max(deg^-(\mathbf{V}))$ and maximum outdegree $max(deg^-(\mathbf{V}))$ are both four and the average

Table 5.3: Parameters and values describing the learned causal graph on the entire machinery dataset. The parameters consist of the number of variables N, the number of learned edges $|\mathbf{E}|$, the average indegree $avg(deg^{-}(\mathbf{V}))$, the maximum indegree $max(deg^{-}(\mathbf{V}))$, the average outdegree $avg(deg^{+}(\mathbf{V}))$ and the maximum outdegree $max(deg^{+}(\mathbf{V}))$.

N	$\lvert\mathbf{E}\rvert$	$avg(deg^{-}(\mathbf{V}))$	$max(deg^{-}(\mathbf{V}))$	$avg(deg^{+}(\mathbf{V}))$	$max(deg^{-}(\mathbf{V}))$
1,903	550	1.21	4	1.12	4

indegree $avg(deg^{-}(\mathbf{V}))$ is slightly higher than the average outdegree $avg(deg^{+}(\mathbf{V}))$ with a factor of 1.21 compared to 1.12.

Concerning the domain knowledge-based edge orientation, we found that nine edges of the 550 edges have a wrong orientation according to the set R. The edges in R stem entirely from ***Rule 1***. Hence, there is no violation of ***Rule 2***. Further, four of these edges have been oriented following Meek's orientation rules [115]. Regarding the unforeseen production stops, eleven unique production stop messages, respectively variables, exist in the data. For seven of these variables, a total of eight causal relationships have been identified.

While there is little error according to the domain knowledge-based edge orientation, full validation of all existing and non-existing edges is out of scope for any domain expert. Additionally, we assume that many nodes without any causal relationship indicate that several variables obtained from the log data have little relevance to the production process. Yet, these variables might introduce additional noise to the model, impacting its overall quality. Hence, fully relying on the learned causal structures within the entire set of variables is not advisable. Thus, the identified causal relationships for the production stop messages should only be considered an indication for further investigation of root causes, for example, through a careful selection of relevant variables by a domain expert, including the identified variables.

Case II (Validation): For the second case, a domain expert selected a subset $\mathbf{V}^{S}$ of 11 variables for which the underlying causal mechanisms are known. In this context, the goal is to derive a relevant and comprehensible sub-problem that can be evaluated through domain expertise. The variables' classification, data type, and description are provided in Table 5.4.

The 11 variables represent three task configuration parameters, two non-critical operational messages, two different production stop messages, and four continuous measurements. Note, the two variables for the measurements, shown in Table 5.4 have a realization during the ramp-up phase Q^{c}_{X} and the manufacturing phase Q^{e}_{X}, with X being either *number* or *speed*. Overall, the variables cover three data types: categorical, binary, or discretized. Note that discretized means a continuous variable has been discretized using binning. The categorical variables of the product dimensions C_x, C_y, C_z have between two to five different categories. The binary variables are either zero or one. A zero indicates that the operational message or production stop did not occur in the corresponding observation. In contrast, one indicates that the operational message or production stop happened in the corresponding observation.

The resulting causal graph is shown in Fig. 5.4. Note, C_x and C_y are causally dependent on each other and have, similar to Q^{c}_{number}, no relationship to any other variable. Therefore, they are omitted in Fig. 5.4. Further, the learned causal relationships suggest an influence of the operational messages O_1, O_2, as well as C_z on the unforeseen production stops S_1, S_2. Both product rotations within the machine and the material's and product's thickness can cause internal machine subsystems to pick up the wrong amount

Table 5.4: The 11 variables contained in the validated dataset, including their classification, their type and a description of its meaning. Note that the bottom two variables Q_{speed} and Q_{number} have two representations one in each task's phase, ramp-up (^{c}) and execution (^{e}).

Variable	Classification	Type	Description
C_x	configuration parameter	categorical	product length
C_y	configuration parameter	categorical	product width
C_z	configuration parameter	categorical	product thickness
O_1	non-critical operational message	binary	product rotated clockwise
O_2	non-critical operational message	binary	product rotated anti-clockwise
S_1	production stop	binary	no material in finishing step
S_2	production stop	binary	too much material in finishing step
$Q_{speed}^{\{e/c\}}$	measurement	discretized	machine speed
$Q_{number}^{\{e/c\}}$	measurement	discretized	number of produced products

of material for the finishing step. Thus, causing a production stop due to no material or too much material in the finishing step. In this case, the relationship between the two production stops is plausible, too. The causal relationship of the product thickness C_z between the machine speeds $Q_{speed}^{\{e/c\}}$, as well as the relationship of the number of produced products Q_{number}^e to the manufacturing speed Q_{speed}^e, is also confirmed. The domain expert questions only the causal relationship of the machine speed during manufacturing Q_{speed}^e to the machine speed during the ramp-up phase Q_{speed}^c. This particular edge is marked by the algorithm as a violation of ***Rule 1*** and is considered to be a wrongly detected edge. We believe the wrong direction to be caused by the discretization of the measurements. Overall, the solution obtained was confirmed by the experts.

The validated learned causal graph is a starting point to support a machine operator to identify causes for the production stops S_1, S_2, e.g., rotations of the product during production O_1, O_2 or the product thickness C_z. In contrast to commonly applied classification methods where the most relevant variables are used for root cause analysis, e.g., see [18] or [35], the knowledge about causal structures not only provides the opportunity to detect sequences of root causes but also distinguishes between associative and causal

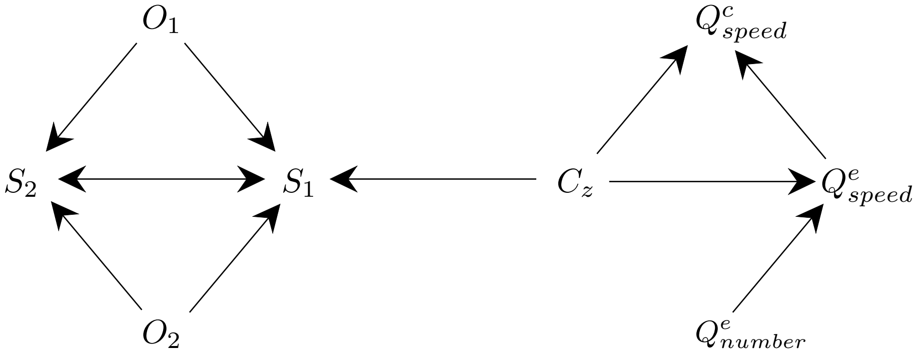


Fig. 5.4: The learned causal graph, resulting from the application of process ***Steps 1*** and ***2***. Note that a subset of 11 variables was selected from the entire machine dataset and that the phase of the measurements, ramp-up (^{c}) or execution (^{e}), is added at the end of the variable.

variables [128]. For example, the product thickness C_z as a common confounder induces an associative dependence of the machine speed Q^c_{speed} and production stops S_1. Hence, the relevance of Q^c_{speed} is increased within the classification approach, although there exists no causal effect of Q^e_{speed} on S_1.

Due to the restrictive selection of variables, sequences of influences beyond the subgraph cannot be identified, e.g., causes for O_1 or O_2, such that domain experts should consider a larger set of relevant variables in practice to investigate influences for O_1 and O_2, too.

5.4.3 *Step 3 (Causal Inference)*: Apply Causal Inference to Understand Key Relationships

The framework of causal graphical models (CGMs) together with the *do*-operator (see Section 1.2 on page page 3) allows for an estimation of causal effects in an experimental regime on the basis of observational data [135]. The *do*-operator provides answers to questions, such as, "*What is the impact of a machine change?*", which might not be affordable in real-world scenarios due to production process interruption or unavailability of an experimental setup.

Furthermore, an examination of causal relationships avoids wrong deductions and decisions based on associational characteristics. For example, consider the following scenario of a domain expert investigating the causes of the production stop S_1, i.e., $S_1 = 1$. In this setting, a domain expert aims to decrease the probability $\mathbb{P}(S_1 = 1)$ by changing the values of possible causes. Let us assume the machine operator considers that the machine speed during manufacturing Q^e_{speed} impacts the occurrence of production stop S_1 as indicated by a classification approach (see ***Case (II)*** of Section 5.4.2). In this setting, the machine operator consults the data, inspecting conditional probabilities of $S_1 = 1$ given any of the five categories of Q^e_{speed} during manufacturing as displayed in Table 5.5. The data indicates that using a machine speed within the range represented by category $k = 1$ yields the lowest probability of an occurrence of the production stop $S_1 = 1$.

In this example, the *do*-operator enables calculating the interventional conditional probabilities of S_1, i.e., $\mathbb{P}(S_1 = 1|do(Q^e_{speed} = k))$, to examine the causal effect of an intervention on Q^e_{speed} to 1. Based on the result, the machine operator could judge if changing the machine speed results in the desired reduction in the occurrence of the production stop $S_1 = 1$. To calculate the respective probabilities, we utilize the `R`-package `causaleffect` [177], which applies rules to determine a formula to calculate the respective probabilities under a given intervention. Note, the `causal.effect` function operates on a Directed Acyclic Graph (DAG). Hence, for the learned graph depicted in Fig. 5.4, we select one of the two represented DAGs of the Markov equivalence class, i.e., extending the CPDAG to a DAG considering $S_1 \rightarrow S_2$. For the case of $S_1 = 1$ the conditional probability given an intervention on S^e_{speed} is given by $\mathbb{P}(S_1 = 1|do(Q^e_{speed} = 1))$. Applying the formula determined by the R-package *causaleffect* when intervening on $Q^e_{speed} = 1$ coincides with the unconditioned probability of $\mathbb{P}(S_1 = 1) = 13.4\%$. Thus, for our example,

Table 5.5: The conditional probabilities for the occurrence of the unique stopper $S_1 = 1$, when either selecting a distinct machine speed during execution $Q^e_{speed} = k$ or not selecting any, represented by unconditioned. The machine speed was discretized into five categories $k = 0, 1, \ldots, 4$ using equal-width binning.

	unconditioned	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$\mathbb{P}(S_1 = 1\|Q^e_{speed} = k)$	13.4%	10.6%	8.6%	15.3%	13.3%	13.3%

changes to the machine speed Q^e_{speed} do not influence S_1 and based on this assumption, the machine operator would have applied impractical changes to the machine setting. Affirmatively, the learned causal graphical model does not contain any causal relationship between the two variables Q^e_{speed} and S_1.

5.5 Discussion

We close this chapter with a summary of our contributions (Section 5.5.1), an examination of limitations, particularly regarding the applicability and transferability to other domains (Section 5.5.2), and future work (Section 5.5.3).

5.5.1 Summary

We proposed a process to learn causal structures in the context of a discrete manufacturing production process, which comes with domain-specific challenges. The causal relationships are learned based on data logged for monitoring the manufacturing machines during operation. Using real-world data from a globally operating machine manufacturer, we showed how to apply all single process steps to derive the causal graphical model, which we utilize to estimate causal effects in an experimental regime. Our example showed how to provide data-driven decision support to assist domain experts in avoiding wrongly assumed influences of unforeseen production stops.

Our proposed process integrates domain knowledge at different steps to increase the quality and interpretability of the results. First, transformation rules are defined to extract sound observations from the log data (see *Challenge II*). Second, domain experts are encouraged to select relevant variables to reduce the search space and noise in high-dimensional settings (see *Challenge I*). To handle mixed data, we suggest discretizing the data (see *Challenge III*) before applying parallel constraint-based causal structure learning algorithms. Further, we extend the rules commonly applied during the algorithm's edge orientation. Note this procedure and the basic methods applied in our approach could also address the causal reasoning from machine log data in similar applications.

5.5.2 Limiations

In the following discussion on limitations, we point out necessary considerations concerning generalizability and limitations that need to be mentioned.

Transferability to Other Domains: We see the potential to apply the proposed process in similar production settings, e.g., automotive production. In these production settings, monitoring systems are in place, and similar messages are logged. Further, using domain knowledge to specify a causal objective in combination with a set of definitions similar to the ones defined in Section 5.4.1, e.g., based on time or location constraints, e.g., by utilizing the `location` information of a `LogMessage`, allows applying the transformation rules to derive sound observational data. To sketch a concrete example from automotive production, consider the causal objective to understand the reasons for defective cars in the context of the car assembly. The car is worked on in different assembly stations within the car assembly at specific points in time [70]. Thus, analogously to our proposed process, time windows can be detected for the car's presence in each station, and all monitoring log messages are mapped accordingly. Hence, the observations can be constructed using the same set of transformation rules, and causal structures can be learned following our proposed process.

Yet, it remains to investigate if the generalized approaches to apply the same aggregation function for a category of messages or a single discretization approach yield acceptable results in these settings. In the context of discretization, we see the potential for an automatic selection of the most appropriate discretization technique for each continuous variable. Such a step requires a validated subset to determine the impact on the learned causal structures and a generalization to the remaining variables, e.g., based on the variable categories. Further, it remains to be investigated if the transformation rules remain applicable in domains in which constraints other than the time dimension are used to derive sound observational data. The applicability of the transformation rules may become a limiting factor for the transferability to other domains.

Methodological Limitations: While the proposed process yields appropriate results in our use case, it is worth examining methodological limitations that need to be considered when applying causal structure learning and causal inference. First, note that the assumptions of methods for causal structure learning are quite restrictive. In this context, it is important to check whether the particular dataset satisfies the required preconditions such as causal sufficiency, i.e., no latent confounding variables. Note that there exists a variety of extensions to the PC algorithm that allow the application of causal structure learning under weakened assumptions, e.g., the FCI algorithm in case of violated causal sufficiency [167]. In this context, it is worth mentioning that our approach may serve as the basis for capturing time-dependent causal effects through the implementation of methodological extensions for time-related causal graphical models with respective algorithms.

Second, besides the theoretical restrictions on the dataset, the accuracy of our approach is strongly influenced by the trustworthiness of the incorporated domain knowledge. In particular, incorrect domain knowledge within the process of causal structure learning from machine log data, see Fig. 5.2, may yield not only unreliable observational data but also wrong causal structures and hence an incorrect causal inference. Moreover, while discretization improves the interpretability of both causal structures and causal effects, a wrong technique may preserve relevant causal relationships.

Third, while adequate within our use case, data quality is a well-known problem in practice and requires an additional step for data cleaning, e.g., the imputation of missing values. In the context of machine log data, changes in both the logging technique and systematic changes within the machine, such as modifications during the observation period, may yield changes within the underlying causal mechanism and hence inconsistent learned causal structures.

In summary, both the implementation of the proposed process for causal structure learning and the interpretation of the results should be made carefully. For more information on the challenges of causal discovery in practice, see the practical guide of Malinsky and Danks [106].

5.5.3 Future Work

In future work, the methodological limitations can be addressed, by considering causal discovery methods with weaker assumptions, e.g., the FCI algorithm [167], through detailed studies on the impact of discretization [29], and investigating model drift concerning systematic changes within machines. Additionally, future work needs to develop a sophisticated validation method to verify entire learned causal models with domain experts. This validation method allows receiving a quantifiable judgment of the correctness of the applied techniques in the given context of the machine log data.

6

Conclusion of Part I

In this chapter, we close Part I by summarizing our results (Section 6.1), point out limitations (Section 6.2), and discuss future work (Section 6.3).

Contribution: Parts of this chapter have previously been published in the papers [68, 67, 54]. For a depiction of the author's contributions and a more detailed discussion on the corresponding limitations and future work, we refer to Chapter 3, Chapter 4, and Chapter 5, respectively.

6.1 Summary

Recap that this first part of this thesis, Part I, was motivated with the goal of providing necessary tool support to improve the evaluation and application of methods for causal discovery in practice (see Section 2.1 on page 25). In this context, we contributed in a threefold manner.

- We proposed an architectural blueprint of a **modular pipeline for causal discovery**, and our reference implementation MPCSL to support data scientists in their research on methods for causal discovery. (see Chapter 3 on page 29).
- We introduced the mixed additive noise model (MANM) to provide a ground truth model for **synthetic data generation** following various distribution models and presented our reference implementation MANM-CS (see Chapter 4 on page 43).
- We demonstrated the process of causal discovery in a **real-world discrete manufacturing scenario** to showcase challenges and requirements and provide concepts for the applicability of causal discovery (see Chapter 5 on page 57).

In the following, we explain the added value of the above contributions regarding improvements in the evaluation and application of methods for causal discovery in practice.

A Modular Pipeline for Causal Discovery: The architectural blueprint of a pipeline for causal structure learning and our reference implementation MPCSL addresses the requirements towards platform independence and modularity while ensuring the comparability and reproducibility of experiments. Therefore, MPCSL provides researchers and practitioners a tool to evaluate and apply methods for causal discovery considering various application scenarios. In this context, a case study on the runtime of the well-known PC algorithms demonstrates the capabilities of MPCSL to evaluate existing implementations concerning their runtime performance in several data characteristics.

Beyond the demonstrated research aspect, we used the developed tool with industry partners from the manufacturing domain. Here, the focus is on comparing the accuracy of learned causal structures with different approaches for causal discovery, as well as different data preparation techniques. In this context, we found that MPCSL covers the strong need for visual support during the selection and preparation of data for causal discovery and when comparing different learned causal structures. For a more detailed summary, we refer to Section 3.5.1 (on page 41).

Synthetic Data Generation for Causal Discovery: We introduced the mixed additive noise model (MANM) to provide a ground truth framework for generating observational data following various distribution models. In this context, our reference implementation MANM-CS provides easy access to researchers and practitioners. In particular, MANM-CS's capabilities not only provide enough opportunities to mimic common benchmarking approaches but also allow for more comprehensive evaluations with varying model complexity. Therefore, MANM-CS, together with MPCSL, provides a framework for benchmarking causal discovery and, hence, supports researchers in the attempt to develop novel algorithms for causal discovery, e.g., to achieve a milder set of assumptions regarding mixed discrete-continuous data. For example, MANM-CS is the basis of the synthetic evaluation of our non-parametric CI test developed for mixed discrete-continuous data (see Part II on page 83). For a more detailed summary, we refer to Section 4.6.1 (on page 56).

Application Scenario in Discrete Manufacturing: We analyzed a real-world use case from a globally operating precision mechanical engineering company to show the applicability of our customized process for causal discovery and causal inference from raw machinery log data. The application of MPCSL and lessons learned in MANM-CS yields a process to learn causal structures from raw machinery log data that could support machine operators, for example, to identify the root causes of unexpected production downtimes. In particular, the machine operator consults the causal structures to determine the variables, which have edges pointing to the production stop message monitored at production downtime. Thus, the search space to remove the production stop is reduced, and time is saved. Note, as the time of experienced machine operators is valuable and limited, such kind of automated data-driven support can be highly beneficial. For a more detailed summary, we refer to Section 5.5.1 (on page 75).

In summary, we contributed a comprehensive toolkit for causal discovery to support researchers and practitioners and showcased its capabilities in a real-world scenario to identify root causes of unexpected product stops.

6.2 Limitations

Although the presented tools provide a comprehensive toolkit, they require a deep understanding of the methods and concepts of causal discovery. For example, using MPCSL within a discrete manufacturing factory presupposes that the machine operator is familiar with causal graphs or the *do*-operator.

Therefore, for efficient integration into the workflow of a machine operator, we suggest integrating the learned causal structures into an existing monitoring solution, see [70]. In this context, we suggest a careful selection of variables when using causal structure learning in practice. On the one hand, to avoid noise, i.e., when using all variables, on the other hand, to avoid missing influences, i.e., when selecting too restrictive.

Further, we suggest visually presenting only the relevant selection of the causal model for the occurring production stop to provide a focus for the machine operator. Extending this monitoring solution with the capability for causal inference further strengthens the support for the machine operator. The integration of causal inference is beneficial for inexperienced machine operators, as it avoids drawing false conclusions (cf. Section 5.4.3).

6.3 Future Work

For future work, we aim to integrate `MAM-CS` into `MPCSL` such that the pipeline envelopes the whole benchmarking pipeline. Furthermore, we consider the application of transformation rules developed in our discrete-manufacturing scenario to other use cases. For more details on future work regarding each contribution we refer to Section 3.5.3 on page 42, Section 4.6.3 on page 56, and Section 5.5.3 on page 76, respectively.

Part II

Causal Discovery from Mixed Discrete-Continuous Data

Testing for conditional independence (CI) is a fundamental task for constraint-based causal discovery but is particularly challenging in mixed discrete-continuous data omnipresent in many real-world scenarios. In this context, inadequate assumptions or discretization of continuous variables reduce the CI test's statistical power, which yields incorrect learned causal structures.

In this part, we tackle **RQ2** "*How to weaken the assumptions of constraint-based causal discovery on data characteristics?*" to improve causal discovery in practice. Therefore, we present a non-parametric CI test leveraging k-nearest neighbors (kNN) methods that are adaptive to mixed discrete-continuous data. In particular, a kNN-based conditional mutual information estimator serves as the test statistic, and the p-value is calculated using a kNN-based local conditional permutation scheme. We prove the CI test's statistical validity and power in mixed discrete-continuous data, which yields consistency when used in constraint-based causal discovery. An extensive evaluation on synthetic and real-world data shows that the proposed CI test outperforms state-of-the-art approaches in the accuracy of CI testing yields more accurate causal structures when used in constraint-based causal discovery, particularly in settings with low sample sizes.

7

Overview: Mixed Discrete-Continuous Data

Conditional Independence (CI) testing is at the core of causal discovery (Section 7.1), but particularly challenging in mixed discrete-continuous data omnipresent in many real-world scenarios (Section 7.2). Therefore, we contribute by providing a data-adaptive CI test for mixed discrete-continuous data (Section 7.3). Further, we outline Part II of this thesis (Section 7.4).

Contribution: Parts of this chapter have previously been published in the paper [69]. The thesis author developed the applied concepts and prepared the original draft. A detailed depiction of the author's contributions is discussed at the beginning of the corresponding chapters Chapter 8, Chapter 9, and Chapter 10, respectively.

7.1 Motivation and Background

Causal discovery has received widespread attention as the knowledge of underlying causal structures improves decision support within many real-world scenarios [44, 168]. For example, in discrete manufacturing, causal discovery is the key to root cause analysis of failures and quality deviations [70, 54].

For a gentle introduction to causal discovery, see Section 1.2 (page 3) and, for an elaborate background on causal discovery, we refer to [168]. However, in short, we provide the necessary notation needed in this part, which can be slightly simplified as we restrict our attention to the conditional independence (CI) testing problem of causal discovery.

Causal structures between a finite set of random variables $\mathbf{V} = \{X, Y, \dots\}$ are encoded in a causal graphical model (CGM) consisting of a directed acyclic graph (DAG) $\mathcal{G}$, and the joint distribution over the variables $\mathbf{V}$, denoted by $P_{\mathbf{V}}$, cf. [130, 168]. In $\mathcal{G}$, a directed edge $X \rightarrow Y$ depicts a direct causal mechanism between the two respective variables X and Y, for $X, Y \in \mathbf{V}$. Causal discovery aims to derive as many underlying causal structures in $\mathcal{G}$ from observational data as possible building upon the coincidence between the causal structures of $\mathcal{G}$ and the CI characteristics of the joint distribution $P_{\mathbf{V}}$ [168]. Therefore, constraint-based methods, such as the well-known PC algorithm, apply CI tests to recover the causal structures, cf. [20]. For instance, if a CI test states the conditional independence of variables X and Y given a (possibly empty) set of variables $Z \subseteq \mathbf{V} \setminus \{X, Y\}$, denoted by $X \perp\!\!\!\perp Y | \, Z$, then there is no edge between X and Y. Constraint-based methods are flexible and exist in various extensions, e.g., to allow for latent variables or cycles [145, 168, 174], or are used for causal feature selection [190]. Hence, they are popular in practice [106].

7.2 Challenges in Practice

In principle, constraint-based methods do not make any assumption on the functional form of causal mechanisms or parameters of the joint distribution. However, they require access to a CI oracle that captures all CI characteristics of $P_{\mathbf{V}}$ such that selecting an appropriate CI test is fundamental and challenging [44, 106]. In practice, the true statistical properties are mostly unknown such that inadequate assumptions, e.g., of parametric CI tests, yield incorrect learned causal structures [168]. For example, the well-known partial Pearson's correlation-based CI test via Fisher's Z transformation assumes that $P_{\mathbf{V}}$ is multivariate Gaussian [4, 75]. Hence, the underlying causal mechanisms are assumed to be linear, and conditional independence cannot be detected if the mechanisms are non-linear. Further, the omnipresence of mixed discrete-continuous data, e.g., continuous quality measurements and discrete failure messages in discrete manufacturing [54], impedes the selection of appropriate CI tests in real-world scenarios [49, 106]. In this case, parametric models that allow for mixed discrete-continuous data usually make further restrictions, such as conditional Gaussian models assuming that discrete variables have discrete parents only [139]. Hence, for simplification in practice, continuous variables are often discretized to use standard CI tests such as Pearson's χ^2 test for discrete data, cf. [54, 65, 110], to the detriment of the accuracy of the learned causal structures [29, 139].

7.3 Contributions

In this work, we propose `mCMIkNN`[7] a data-adaptive CI test for mixed discrete-continuous data and its application to causal discovery. In particular, we contribute the following:

- We propose a kNN-based local conditional permutation (CP) scheme to derive a non-parametric CI test, called `mCMIkNN`, building upon a kNN-based CMI estimator as a test statistic.
- We provide theoretical results on the CI test's validity and power. In particular, we prove that `mCMIkNN` is able to control type I and type II errors.
- We show that `mCMIkNN` allows for consistent estimation of causal structures when used in constraint-based causal discovery.
- An extensive evaluation on synthetic and real-world data shows that `mCMIkNN` outperforms state-of-the-art competitors, particularly for low sample sizes.
- We show that `mCMIkNN` improves the accuracy of causal discovery in a real-world discrete manufacturing scenario.

7.4 Outline of Part II

The remainder of this part is structured as follows. In Chapter 8 (page 87), we examine the problem of CI testing and related work, provide background on kNN-based CMI estimation, and introduce `mCMIkNN` as well as prove theoretical results. In Chapter 9 (page 103), we empirically evaluate the accuracy of `mCMIkNN` in CI testing and causal discovery compared to state-of-the-art approaches. In Chapter 10 (page 117), we apply `mCMIkNN` in a real-world discrete manufacturing scenario. In Chapter 11 (page 123), we conclude our work and discuss limitations and future research directions.

[7] https://github.com/hpi-epic/mCMIkNN

8

Non-Parametric CI Testing

In this chapter, we provide a formalization of the conditional independence (CI) testing problem together with existing fundamental limits of CI testing (Section 8.1) before considering related work on CI testing for mixed discrete-continuous data (Section 8.2). Then, we provide the necessary background on kNN-based conditional mutual information (CMI) estimation (Section 8.3). On this basis, we introduce our approach for kNN-based CI testing in mixed discrete-continuous data, called `mCMIkNN`, and prove theoretical results (Section 8.4).

Contribution: Parts of this chapter have previously been published in the paper [69]. The thesis author developed all statistical concepts, implemented the methods, and prepared the original draft of the paper. Christopher Hagedorn supported the implementation of `mCMIkNN`. The coauthors improved the paper's material and its presentation.

8.1 Problem Description and Fundamental Limits

In this section, we provide a formalization of the conditional independence (CI) testing problem (Section 8.1.1) and examine existing fundamental limits of CI testing (Section 8.1.2).

8.1.1 Problem Description

Let $(\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}, \mathcal{B}, P_{XYZ})$ be a probability space defined on the metric space $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ with dimensionality $d_X + d_Y + d_Z$, equipped with the Borel σ-algebra $\mathcal{B}$, and a regular joint probability measure P_{XYZ}. Hence, we assume that d_X, d_Y, and d_Z-dimensional random variables X, Y, and Z take values in $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ according to the marginal mixed discrete-continuous probability distributions P_X, P_Y, and P_Z. I.e., single variables in X, Y, or Z may follow a discrete, a continuous, or a mixture distribution.

We consider the problem of testing the CI of two random vectors X and Y given a (possibly empty) random vector Z sampled according to the mixed discrete-continuous probability distribution P_{XYZ}, i.e., testing the null hypothesis of CI $H_0 : X \perp\!\!\!\perp Y \,|\, Z$ against the alternative hypothesis of dependence $H_1 : X \perp\!\!\!\perp Y \,|\, Z$. Therefore, let $(x_i, y_i, z_i)_{i=1}^n$ be n i.i.d. observations sampled from P_{XYZ} such that we aim to derive a CI test $\varPhi_n : \mathcal{X}^n \times \mathcal{Y}^n \times \mathcal{Z}^n \times [0,1] \to \{0,1\}$ that rejects H_0 if $\varPhi_n = 1$ given a nominal level $\alpha \in [0,1]$.

8.1.2 Fundamental Limits of CI Testing

The general problem of CI testing is extensively studied, as it is a fundamental concept beyond its application in constraint-based causal discovery [28].

In this context, it is necessary to note that Shah and Peters [162] provided a *no-free lunch theorem* for CI that, given a continuously distributed conditioning set Z, it is impossible to derive a CI test that can control the type I error, via for instance a permutation scheme, and has nontrivial power without additional restrictions. But, under the restriction that the conditional distribution $P_{X|Z}$ is known or can be approximated sufficiently, conditional permutation (CP) tests can calibrate a test statistic guaranteeing a controlled type I error [7]. Further, the recent work of Kim et al. [78] shows that the problem of CI testing is more generally determined by the probability of observing *collisions* in Z.

8.2 Related Work on CI Testing

We consider the problem of conditional independence (CI) testing in mixed discrete-continuous data and its application in causal discovery. In this context, constraint-based methods require CI tests that

- **(R1)** yield accurate CI decisions in mixed discrete-continuous data and
- **(R2)** are computationally feasible as they may be applied hundreds of times.

Generally, testing for CI in mixed discrete-continuous data can be categorized into discretization-based (Section 8.2.1), parametric (Section 8.2.2), and non-parametric approaches (Section 8.2.3).

8.2.1 Discretization-Based Approaches

As CI tests for discrete variables are well-studied, continuous variables are often discretized, cf. [65, 110]. In this context, commonly used CI tests for discrete data are Pearson's $\mathcal{X}^2$ and likelihood ratio tests [36, 132, 168]. Although discretization simplifies the testing problem, the resulting information loss yields a decreased accuracy of CI decisions [29, 139], see **(R1)**.

8.2.2 Parametric CI Testing

Postulating an underlying parametric functional model allows for a regression-based characterization of CI that can be used to construct valid CI tests. Examples are well-known likelihood ratio tests, e.g., assuming conditional Gaussianity (CG) [2, 157] or using multinomial logistic regression models [180].

Another stream of research focuses on Copula models to examine CI characteristics in mixed discrete-continuous data, where variables are assumed to be induced by latent Gaussian variables such that CI can be determined by examining the correlation matrix of the latent variables model [24, 25]. As these approaches require that the postulated parametric models hold, they may yield invalid CI decisions if assumptions are inaccurate [168], cf. **(R1)**.

8.2.3 Non-Parametric CI Testing

Non-parametric CI testing faces the twofold challenge to, first, derive a test statistic from observational data without parametric assumptions, and second, derive the p-value given that the test statistic's distribution under H_0 may be unknown.

In continuous data, a wide range of methods is used for non-parametric CI testing, as reviewed by Li and Fan [94]. For example, kernel-based approaches, such as `KCIT` [195], test for vanishing correlations within Reproducing Kernel Hilbert Spaces (RKHS). Another example is `CMIknn` from Runge [148], which uses a k-nearest neighbors (kNN)-based estimator to test for a vanishing conditional mutual information (CMI) in combination with a local permutation scheme.

The recent emergence of non-parametric CMI estimators for mixed discrete-continuous data provides the basis for new approaches to non-parametric CI testing. For example, the construction of adaptive histograms derived following the minimum description length (MDL) principle allows for estimating CMI from mixed discrete-continuous data [14, 107, 112, 191]. In this case, CMI can be estimated via discrete plug-in estimators as the data is adaptively discretized according to the histogram with minimal MDL. Hence, the estimated test statistic follows the common $\mathcal{X}^2$ distribution, which allows for derivation via Pearson's $\mathcal{X}^2$ test, which we refer to as aHisχ^2, see [112]. However, MDL approaches suffer from their worst-case computational complexity and weaknesses regarding a low number of samples, see **(R2)**.

Another approach for non-parametric CMI estimation builds upon kNN methods, which are well-studied in continuous data, cf. [41, 82, 83], and have recently been applied to mixed discrete-continuous data [43, 117]. As the asymptotic distribution of kNN-based estimators is unclear, it remains to show that they can be used as a test statistic for a valid CI. In this context, it is worth noticing that permutation tests yield more robust constraint-based causal discovery than asymptotic CI tests, particularly for small sample sizes [181], see **(R1)**.

In this work, we combine a kNN-based CMI estimator with our novel kNN-based local conditional permutation (CP) scheme (similar to Runge [148], which is restricted to the continuous case), and additionally provide theoretical results on the test's validity and power as well its asymptotic consistency when used in constraint-based causal discovery.

8.3 Background on kNN-Based CMI Estimation

In this section, we provide information on kNN-based CMI estimation for mixed discrete-continuous data (Section 8.3.1). Further, we introduce an algorithmic description of the estimator (Section 8.3.2) and recap theoretical results (Section 8.3.3).

8.3.1 Introduction to CMI Estimation

A commonly used test statistic is the conditional mutual information (CMI) $I(X;Y|Z)$ as it provides a general measure of variables' conditional independence (CI), i.e., it holds that $I(X;Y|Z) = 0$ if and only if $X \perp\!\!\!\perp Y \mid Z$, see [43, 45, 148]. Generally, $I(X;Y|Z)$ is defined as

$$I(X;Y|Z) = \int \log\left(\frac{dP_{XY|Z}}{d\big(P_{X|Z} \times P_{Y|Z}\big)}\right) dP_{XYZ}, \tag{8.1}$$

where $\frac{dP_{XY|Z}}{d\left(P_{X|Z} \times P_{Y|Z}\right)}$ is the Radon-Nikodym derivative of the joint conditional measure, $P_{XY|Z}$, with respect to the product of the marginal conditional measures, $P_{X|Z} \times P_{Y|Z}$. Note the non-singularity of P_{XYZ} ensures the existence of a product reference measure and that the Radon-Nikodym derivative is well-defined [117, Lemma 2.1, Theorem 2.2].

Although well-defined, estimating CMI $I(X;Y|Z)$ from mixed discrete-continuous data is a particularly hard challenge [43, 112, 117]. Generally, CMI estimation can be tackled by expressing $I(X;Y|Z)$ in terms of Shannon entropies, i.e., $I(X;Y|Z) = H(X,Y,Z) - H(X,Z) - H(Y,Z) + H(Z)$ with Shannon entropy $H(W)$ for all cases $W = XYZ, XZ, YZ, Z$, respectively, cf. [45, 112, 117]. In the continuous case, the KSG technique from Kraskov et al. [83] estimates the Shannon entropy $H(W)$ locally for every sample $(w_i)_{i=1}^n$ where $w_i \sim P_W$, for $W = XYZ, XZ, YZ, Z$, i.e., estimating $H(W)$ via

$\widehat{H}_n(W) = -\sum_{i=1}^{n} \log \widehat{f_W(w_i)}$ by considering the k-nearest neighbors within the ℓ_∞-norm for every sample $i = 1, ..., n$ to locally estimate the density f_W density of $W = XYZ, XZ, YZ, Z$, respectively, cf. [45, 112, 117]. For mixed discrete-continuous data, there is a non-zero probability that the kNN distance is zero for some samples i. In this case, Gao et al. [43] extended the KSG technique by fixing the radius and using a plug-in estimator that differentiates between mixed, continuous, and discrete components.

Recently, Mesner and Shalizi [117] extended this idea to derive a consistent estimator for CMI in the mixed discrete-continuous case.

8.3.2 Algorithm for kNN-Based CMI Estimation

Algorithm 1 provides an algorithmic description of the theoretically examined estimator $\hat{I}_n(X; Y|Z)$ developed by Mesner and Shalizi [117].

The basic idea is to take the mean of Shannon entropies estimated locally for each sample $i = 1, ..., n$ considering samples $j \neq i$, $j = 1, ..., n$, that are close to i according to the ℓ_∞-norm, i.e., under consideration of the respective sample distance $d_{i,j}(w) := \|(w_i) - (w_j)\|_\infty$, $i, j = 1, ..., n$, of $w = (w_i)_{i=1}^n$ for all cases $w = xyz, xy, yz, z$ (see Algorithm 1, line 1). In this context, fixation of a k-nearest neighbors (kNN) radius ρ_i used for local estimation of Shannon entropies yields a consistent global estimator.

Algorithm 1 kNN-Based CMI Estimator [117]

```
Input: Samples $(x, y, z) := (x_i, y_i, z_i)_{i=1}^n$, and kNN-parameter $k_{CMI}$
Output: The estimated value $\hat{I}_n(x; y|z)$ of the CMI $I(X; Y|Z)$
 1: Let $d_{i,j}(w) := \|(w_i) - (w_j)\|_\infty$ for $w \subseteq (x, y, z)$, $i, j = 1, \dots, n$
 2: for $i = 1, \dots, n$ do
 3:     $\rho_i :=$ the $k_{CMI}$-smallest distance in $\{d_{i,j}(x, y, z), j \neq i\}$        ▷ Adapt $k_{CMI}$ acc. $\rho_i$
 4:     $\tilde{k}_i := |\{(x_j, y_j, z_j) : d_{i,j}(x, y, z) \leq \rho_i, j \neq i\}|$
 5:     $n_{xz,i} := |\{(x_j, z_j) : d_{i,j}(x, z) \leq \rho_i, j \neq i\}|$        ▷ Local estimates
 6:     $n_{yz,i} := |\{(y_j, z_j) : d_{i,j}(y, z) \leq \rho_i, j \neq i\}|$
 7:     $n_{z,i} := |\{(z_j : d_{i,j}(z) \leq \rho_i, j \neq i\}|$
 8:     $\xi_i := \psi(\tilde{k}_i) - \psi(n_{xz,i}) - \psi(n_{yz,i}) + \psi(n_{z,i})$
 9: end for
10: $\hat{I}_n(x; y|z) = \frac{1}{n} \sum_{i=1}^{n} \xi_i$        ▷ Global CMI estimation
11: return $\max(\hat{I}_n(x; y|z), 0)$
```

Therefore, for each sample $i = 1, \dots, n$, let ρ_i be the smallest distance between (x_i, y_i, z_i) and the k_{CMI}-nearest sample (x_j, y_j, z_j), $j \neq i, j = 1, \dots, n$, and replace k_{CMI} with $\tilde{k}_i$, the number of samples whose distance to (x_i, y_i, z_i) is smaller or equal to ρ_i (see Algorithm 1, line 3-4). For discrete or mixed discrete-continuous samples $(x_i, y_i, z_i)_{i=1}^n$, it holds that $\rho_i = 0$, and there may be more samples than k_{CMI} samples with zero distance. In this case, adapting the number of considered samples $\tilde{k}_i$ to all samples with zero distance prevents undercounting, which, otherwise, yields a bias of the CMI estimator, see [117]. In case of continuous samples $(x_i, y_i, z_i)_{i=1}^n$, there are exactly $\tilde{k}_i = k_{CMI}$ samples within the k_{CMI}-nearest distance with probability 1. The next step estimates the Shannon entropies required by the $3H$-principle locally for each sample i, $i = 1, \dots, n$. Therefore, let $n_{xz,i}, n_{yz,i}$, and $n_{z,i}$ be the numbers of $\tilde{k}_i$-nearest samples within the distance of ρ_i in the respective subspace XZ, YZ, and Z (see Algorithm 1, lines 5-7). Fixing the local kNN distance ρ_i, using the ℓ_∞-norm, simplifies the local estimation as most relevant terms for CMI estimation using the $3H$-principle cancel out, i.e., $\xi_i := -\widehat{f_{XYZ}(x_i, y_i, z_i)} + \widehat{f_{XZ}(x_i, z_i)} + \widehat{f_{YZ}(y_i, z_i)} - \widehat{f_Z(z_i)} = \psi(\tilde{k}_i) - \psi(n_{xz,i}) - \psi(n_{yz,i}) + \psi(n_{z,i})$, with digamma function ψ (see Algorithm 1, line 8) [43, 117].

Then, the global CMI estimate $\hat{I}_n(x;y|z)$ is the average of the local CMI estimates ξ_i of each sample $(x_i, y_i, z_i)_{i=1}^n$, and the positive part is returned, as CMI or mutual information (MI) are non-negative (see Algorithm 1, line 10-11).

8.3.3 Properties of kNN-based CMI Estimation

We recap the theoretical results of $\hat{I}_n(X, Y|Z)$ proved by Mesner and Shalizi [117], and we infer its consistency. Under mild assumptions, $\hat{I}_n(x;y|z)$ is asymptotically unbiased, see [117, Thm. 3.1].

Corollary 1 (Asymptotic-Unbiasedness of $\hat{I}_n(x;y|z)$ [117, Thm. 3.1]). *Let $(x_i, y_i, z_i)_{i=1}^n$ be i.i.d. samples from P_{XYZ}. Assume*

(A1) $P_{XY|Z}$ is non-singular such that $f \equiv \frac{dP_{XY|Z}}{d(P_{X|Z} \times P_{Y|Z})}$ is well-defined, and assume, for some $C > 0$, $f(x, y, z) < C$ for all $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$;

(A2) $\{(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} : P_{XYZ}((x, y, z)) > 0\}$ countable and nowhere dense in $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$;

(A3) $k_{CMI} = k_{CMI,n} \to \infty$ and $\frac{k_{CMI,n}}{n} \to 0$ as $n \to \infty$;

then $\mathbb{E}_{P_{XYZ}}\left[\hat{I}_n(x;y|z)\right] \to I(X;Y|Z)$ as $n \to \infty$.

While *(A1)* seems rather technical, checking for non-singularity is helpful for data analysis by checking sufficient conditions. Given non-singularity, assumptions *(A2)* and *(A3)* are satisfied whenever P_{XYZ} is (i) (finitely) discrete, (ii) continuous, (iii) some dimensions are (countably) discrete and some are continuous, and (iv) a mixture of the previous cases, which covers most real-world data. For more details on the assumptions, see Section 8.4.5.

Next, we prove that the $\hat{I}_n(X;Y|Z)$, as described in Algorithm 1, is an asymptotic consistent estimator of $I(X;Y|Z)$.

Corollary 2 (Consistency of $\hat{I}_n(x;y|z)$). *Let $(x_i, y_i, z_i)_{i=1}^n$ be i.i.d. samples from P_{XYZ} and assume (A1)-(A3) of Corollary 1 hold. Then, for all $\epsilon > 0$,*

$$\lim_{n\to\infty} \mathbb{P}_{P_{XYZ}}\left(\left|\hat{I}_n(x;y|z) - I(X;Y|Z)\right| > \epsilon\right) = 0. \tag{8.2}$$

Proof. Recap that $\hat{I}_n(x;y|z)$ has asymptotic vanishing variance [117, Thm. 3.2], i.e., $\lim_{n\to\infty} \mathrm{Var}(\hat{I}_n(x;y|z)) = 0$, and is asymptotically unbiased, see Corollary 1 or [117, Thm. 3.1]. The consistency of $\hat{I}_n(x;y|z)$ follows from Chebyshev's inequality. □

Therefore, the kNN-based estimator described in Algorithm 1 serves as a valid test statistic for $H_0 : X \perp\!\!\!\perp Y \mid Z$ vs. $H_1 : X \not\perp\!\!\!\perp Y \mid Z$.

Note that $\hat{I}_n(x;y|z)$ is biased towards zero for high-dimensional data with fixed sample size, i.e., it suffers from the curse of dimensionality, see [117, Thm. 3.3].

Corollary 3 (Dimensionality-Biasedness of $\hat{I}_n(x;y|z)$ [117, Thm. 3.3]). *Let $(x_i, y_i, z_i)_{i=1}^n$ be i.i.d. samples from P_{XYZ} and assume (A1)-(A3) of Corollary 1 hold, if the entropy rate of Z is nonzero, i.e., $\lim_{d_Z\to\infty} \frac{1}{d_Z} H(Z) \neq 0$, then, for fixed dimensions d_X and d_Y, $\mathbb{P}_{P_{XYZ}}\left(\hat{I}_n(x;y|z) = 0\right) \to 1$ as $d_Z \to \infty$.*

Hence, even with asymptotic consistency, one must pay attention when estimating $\hat{I}_n(X;Y|Z)$ in high-dimensional settings, particularly for low sample sizes.

8.4 mCMIkNN: A kNN-Based Non-Parametric CI Test

In this section, we recap the concept of conditional permutation (CP) schemes for conditional independence (CI) testing (Section 8.4.1). Then, we introduce our approach for kNN-based CI testing in mixed discrete-continuous data, called mCMIkNN (Section 8.4.2). Further, we prove that mCMIkNN can control type I and type II errors (Section 8.4.3). Moreover, we examine mCMIkNN-based causal discovery and prove its consistency (Section 8.4.4). Finally, we recap the required assumptions and their implication for application (Section 8.4.5).

8.4.1 Introduction to Conditional Permutation Schemes

Using permutation schemes for non-parametric independence testing between two variables X and Y has a long history in statistics, cf. [11, 61, 92].

The basic idea is to compare an appropriate test statistic for independence calculated from the original samples $(x_i, y_i)_{i=1}^n$ against the test statistics calculated M_{perm} times from samples $(x_{\pi_m(i)}, y_i)_{i=1}^n$ for a permutation π_m of $\{1, \ldots, n\}$, $m = 1, \ldots, M_{perm}$, i.e., where samples of X are randomly permuted such that $H_0 : X \perp\!\!\!\perp Y$ holds. In the discrete case, a permutation scheme to test for conditional independence (CI), i.e., for $H_0 : X \perp\!\!\!\perp Y \mid Z$, can be achieved by permuting X for each realization $Z = z$ to utilize the unconditional $X \perp\!\!\!\perp Y \mid Z = z$. In contrast, testing for CI in continuous or mixed discrete-continuous data is more challenging [162], as simply permuting X without considering the confounding effect of Z may yield very different marginal distributions, hence, suffers in type I error control [7, 78].

Therefore, conditional permutation (CP) schemes aim to compare a test statistic estimated from the original data $(x_i, y_i, z_i)_{i=1}^n$, with test statistics estimated from, conditionally on Z, permuted samples $(x_{\pi_m(i)}, y_i, z_i)_{i=1}^n$, $m = 1, ..., M_{perm}$ to ensure $H_0 : X \perp\!\!\!\perp Y \mid Z$. Then, the $M_{perm} + 1$ samples $(x_i, y_i, z_i)_{i=1}^n$ and $(x_{\pi_m(i)}, y_i, z_i)_{i=1}^n$, $m = 1, ..., M_{perm}$ are exchangeable under H_0, i.e., are drawn with replacement such that the p-value can be calculated in line with common Monte Carlo simulations [7, 78]. This requires either an approximation of $P_{X|Z}$ either based upon model assumptions to simulate $P_{X|Z}$ [7], or using an adaptive binning strategy of Z such that permutations can be drawn for each binned realization $Z = z$ [78] (which are both focusing on the continuous case).

To provide a data-adaptive approach valid in mixed discrete-continuous data without too restrictive assumptions, see *(R1)*, which is computationally feasible, see *(R2)*, we propose a local CP scheme leveraging ideas of kNN-based methods, see Section 8.3. In particular, our local CP scheme draws samples $(x_{\pi_m(i)}, y_i, z_i)_{i=1}^n$ such that (I) the marginal distributions are preserved, and (II) x_i is replaced by $x_{\pi_m(i)}$ only locally regarding the k_{perm}-nearest distance σ_i in the space of Z. Intuitively, the idea is similar to common conditional permutation schemes in the discrete case, where entries of the variable X are permuted for each realization $Z = z$, but considering local permutations regarding the neighborhood of $Z = z$.

8.4.2 Algorithm for kNN-Based CI Testing

Algorithm 2 gives an algorithmic description of our kNN-based local CP scheme for non-parametric CI testing in mixed discrete-continuous data.

First, the sample CMI value $\hat{I}_n := \hat{I}_n(x; y|z)$ is estimated from the original samples via Algorithm 1 with parameter k_{CMI} (see Algorithm 2, line 1). To receive local conditional permutations for each sample $(x_i, y_i, z_i)_{i=1}^n$, the k_{perm}-nearest neighbor distance σ_i w.r.t. the ℓ_∞-norm of the subspace of Z is considered. Hence, $\tilde{\mathbf{z}}_i$ is the respective set of indices $j \neq i$, $j = 1, ..., n$ of points with distance smaller or equal to σ_i in the subspace

Algorithm 2 `mCMIkNN`: kNN-Based Non-Parametric CI Test

```
Input: Samples $(x, y, z) := (x_i, y_i, z_i)_{i=1}^n$, and parameters $k_{CMI}, k_{perm}$, and $M_{perm}$
Output: The estimated $p$-value $p_{perm,n}$ for $H_0 : X \perp\!\!\!\perp Y \mid Z$
 1: $\hat{I}_n := \hat{I}_n(x; y|z)$
 2: for $i = 1, \ldots, n$ do                    ▷ Neighbors within $k_{perm}$NN-distance $\sigma_i$ in $Z$
 3:     $\sigma_i := k_{perm}$ smallest distance in $\{\|(z_i) - (z_j)\|_\infty, j \neq i$, for $i, j = 1, ..., n\}$
 4:     $\tilde{\mathbf{z}}_i := \{j : \|(z_i) - (z_j)\|_\infty \leq \sigma_i, j \neq i\}$
 5: end for
 6: for $m = 1, \ldots, M_{perm}$ do                                        ▷ Local CP scheme
 7:     $\pi_m^i :=$ permutation of $\tilde{\mathbf{z}}_i, i = 1, \ldots, n$
 8:     $\pi_m := \pi_m^1 \circ \cdots \circ \pi_m^n$;
 9:     $\hat{I}_n^{(m)} := \hat{I}_n\left(x^{(m)}; y|z\right)$ where $x^{(m)} := (x_{\pi_m(i)})_{i=1}^n$
10: end for
11: $p_{perm,n} := \frac{1}{1+M_{perm}}\left(1 + \sum_{m=1}^{M_{perm}} \mathbb{1}\{\hat{I}_n^{(m)} \geq \hat{I}_n\}\right)$     ▷ Monte Carlo $p$-value
12: return $p_{perm,n}$
```

of Z (see Algorithm 2, lines 3-4). According to a Monte Carlo procedure, samples are permuted M_{perm} times (see Algorithm 2, line 6). For each $m = 1, \ldots, M_{perm}$, the local conditional permutation π_m^i, $i = 1, \ldots, n$, is a random permutation of the index set of $\tilde{\mathbf{z}}_i$ such that the global permutation scheme π_m of the samples' index set $\{1, \ldots, n\}$ is achieved by concatenating all local permutations, i.e., $\pi_m := \pi_m^1 \circ ... \circ \pi_m^n$ (see Algorithm 2, lines 7-8). In the case of discrete data, $\tilde{\mathbf{z}}_i$ contains all indices of samples j with distance $\rho_i = 0$ to z_i, i.e., the permutation scheme coincides with discrete permutation tests where permutations are considered according to $Z = z_i$. In the continuous case, $\tilde{\mathbf{z}}_i$ contains exactly the, in space Z, k_{perm}-nearest neighbors' indices and the global permutation scheme approximates $P_{X|Z=z_i}$ locally within k_{perm}-NN distance σ_i of z_i. Therefore, local conditional permuted samples $(x_{\pi_m(i)}, y_i, z_i)$ are drawn by shuffling the values of x_i according to π_m and respective CMI values $\hat{I}_n^{(m)} := \hat{I}_n\left(x^{(m)}; y|z\right)$ are estimated using Algorithm 1 (see Algorithm 2, line 9). Hence, by construction, $(x_{\pi_m(i)}, y_i, z_i)$ are drawn under $H_0 : X \perp\!\!\!\perp Y \mid Z$ such that the p-value $p_{perm,n}$ can be calculated according to a Monte Carlo scheme comparing the samples' CMI value $\hat{I}_n$ with the H_0 CMI values $\hat{I}_n^{(m)}$ (see Algorithm 2, line 11).

We define the CI test `mCMIkNN` as $\Phi_{perm,n} := \mathbb{1}\{p_{perm,n} \leq \alpha\}$ for the $p_{perm,n}$ returned by Algorithm 2 and, hence, reject $H_0 : X \perp\!\!\!\perp Y \mid Z$ if $\Phi_n = 1$.

The computational complexity of `mCMIkNN` is determined by the kNN searches in Algorithm 1 and Algorithm 2, which is implemented in $\mathcal{O}(n \times log(n))$ using k-d-trees. For more details on assumptions, parameters, and computational complexity, see Section 8.4.5.

8.4.3 Properties of `mCMIkNN`

The following two theorems show that `mCMIkNN` is valid, i.e., can control type I errors as shown in Theorem 1, and has nontrivial power, i.e., can control type II errors as shown in Theorem 2 (on page 97).

Theorem 1 (Validity: Type I Error Control of $\Phi_{perm,n}$).
Let $(x_i, y_i, z_i)_{i=1}^n$ be i.i.d. samples from P_{XYZ}, and assume (A1) - (A3), and

(A4) $k_{perm} = k_{perm,n} \to \infty$ and $\frac{k_{perm,n}}{n} \to 0$ as $n \to \infty$,

hold, then $\Phi_{perm,n}$ with p-value estimated according to Algorithm 2 can control the type I error, i.e., for any desired nominal value $\alpha \in [0, 1]$, when H_0 is true, then

$$\lim_{n \to \infty} \mathbb{E}_{P_{XYZ}}[\Phi_{perm,n}] \leq \alpha. \tag{8.3}$$

Note that this holds true independent of the test statistic $T_n : \mathcal{X}^n \times \mathcal{Y}^n \times \mathcal{Z}^n \to \mathbb{R}$. The idea of the proof is to bound the type I error using the total variation distance between the samples' conditional distribution $P^n_{X|Z}$ and the conditional distribution $\widetilde{P}^n_{X|Z}$, approximated by the local CP scheme to simulate H_0 and show that it vanishes for $n \to \infty$.

Proof. First, we use similar arguments as in the proof of [7, Thm. 4] to show that, under H_0, the type I error of $\Phi_{perm,n}$ (see Algorithm 2) can be bounded in the finite case by the total variation distance of $P^n_{X|Z}$ and $\widetilde{P}^n_{X|Z}$ simulated with the local CP scheme. Note that *(A1)* and *(A2)* ensure the well-definiteness of all marginal distributions for a regular probability space throughout this proof.

Therefore, given that $P_{XY|Z}$ is non-singular, *(A1)*, regularity ensures that $P_{XY|Z} \ll P_{X|Z} \times P_{Y|Z}$ such that, under H_0, we have $P_{XYZ} \equiv P_{X|Z} \times P_{Y|Z} \times P_Z$, see [117, Thm. 2.2]. Further, we define the simulated product measure $\widetilde{P}_{XYZ} = \widetilde{P}_{X|Z} \times P_{Y|Z} \times P_Z$, where $P_{Y|Z}$ and P_Z are the marginals of $P_{XY|Z}$ and P_{XYZ}, respectively, and where $\widetilde{P}_{X|Z}$ is the approximated conditional probability distribution of permuted samples in Algorithm 2. In this context, note that, in the finite case, the distribution of $\widetilde{P}_{X|Z}$ depends on the distribution of the observed samples $(x_i, z_i)_{i=1}^n$. We write $\widetilde{P}^n_{X|Z}$ and $P^n_{X|Z}$ to denote the samples' distribution in the finite case, respectively. In particular, let $\mathcal{S}_n$ denote the set of permutations on the indices $\{1, \dots, n\}$ such that, for a permutation $\pi_m \in \mathcal{S}_n$ sampled according to the local CP scheme in Algorithm 2, $\widetilde{P}^n_{X|Z}$ denotes the samples' conditional distribution where samples of X are permuted according to $\pi_m \in \mathcal{S}_n$, i.e., where $x^{(m)} = (x_{\pi_m(i)})_{i=1}^n$ with $(x_{\pi_m(i)}, z_i) \sim \widetilde{P}^n_{X|Z=z_i}$, $m = 1, \dots, M_{perm}$.

Let $\tilde{x} = (\tilde{x}_i)_{i=1}^n$ be drawn from $\widetilde{P}_{X|Z}$, and let M_{perm} permutations $\tilde{x}^{(m)} = (\tilde{x}_{\pi_m(i)})_{i=1}^n$, $m = 1, \dots, M_{perm}$ be drawn from the local CP scheme of Algorithm 1 sampled from $\tilde{x}$ instead of the true values in x.

Now, we define

$$A_\alpha := \left\{ (x,y,z), (x^{(1)},y,z), ..., (x^{(M_{perm})},y,z) : \frac{1 + \sum_{m=1}^{M_{perm}} \mathbb{I}\{\hat{I}_n^{(m)} \geq \hat{I}_n\}}{1 + M_{perm}} \leq \alpha \right\},$$

where $\hat{I}_n = \hat{I}_n(x;y|z)$ and $\hat{I}_n^{(m)} = \hat{I}_n(x^{(m)};y|z)$ i.e., the set where $p_{perm,n} \leq \alpha$. Then, by definition of A_α, we have that

$$\begin{aligned}
&\mathbb{E}_{P_{XYZ}}[\Phi_{perm,n}] \\
&\quad = \mathbb{P}_{P_{XYZ}}\Big((x,y,z), (x^{(1)},y,z), ..., (x^{(M_{perm})},y,z) \in A_\alpha\Big) \\
&\quad \leq \mathbb{P}_{P_{XYZ}}\Big((\tilde{x},y,z), (\tilde{x}^{(1)},y,z), ..., (\tilde{x}^{(M_{perm})},y,z) \in A_\alpha\Big) + \mathcal{D}_{TV}\Big(P^n_{XYZ}, \widetilde{P}^n_{XYZ}\Big),
\end{aligned}$$

with total variation distance $\mathcal{D}_{TV}\Big(P^n_{XYZ}, \widetilde{P}^n_{XYZ}\Big) = \sup_{A \in \mathcal{B}} |P^n_{XYZ}(A) - \widetilde{P}^n_{XYZ}(A)|$. Since $(\tilde{x}^{(1)},y,z), ..., (\tilde{x}^{(M_{perm})},y,z)$ are clearly i.i.d. sampled according to $\widetilde{P}_{XYZ}$, and are therefore exchangeable, by definition of A_α, we must have

$$\mathbb{P}_{P_{XYZ}}\Big((\tilde{x},y,z), (\tilde{x}^{(1)},y,z), ..., (\tilde{x}^{(M_{perm})},y,z) \in A_\alpha\Big) \leq \alpha.$$

Further, by construction of $\widetilde{P}_{XYZ}$, it holds that

$$\mathcal{D}_{TV}\Big(P^n_{XYZ}, \widetilde{P}^n_{XYZ}\Big) = \mathcal{D}_{TV}\Big(P^n_{X|Z}, \widetilde{P}^n_{X|Z}\Big).$$

Hence, $\mathbb{E}_{P_{XYZ}}[\Phi_{perm,n}] \leq \alpha + \mathcal{D}_{TV}\left(P^n_{X|Z}, \widetilde{P}^n_{X|Z}\right)$.

Next, we show that $\mathcal{D}_{TV}\left(P^n_{X|Z}, \widetilde{P}^n_{X|Z}\right)$ diminishes for $\frac{k_{perm,n}}{n} \to 0$ as $n \to \infty$. In this context, we relate the total variation distance to the Kullback-Leibler divergence using Pinsker's inequality, namely

$$\mathcal{D}_{TV}\left(P^n_{X|Z}, \widetilde{P}^n_{X|Z}\right) \leq \sqrt{\frac{1}{2}\mathcal{D}_{KL}\left(P^n_{X|Z}\middle\|\widetilde{P}^n_{X|Z}\right)}, \tag{8.4}$$

where $\mathcal{D}_{KL}(P^n_{X|Z}\|\widetilde{P}^n_{X|Z}) = \displaystyle\int \log(\frac{dP^n_{X|Z}}{d\widetilde{P}^n_{X|Z}})dP^n_{X|Z}$ denotes the Kullback-Leibler divergence. Notice that, by construction, $P^n_{X|Z} \ll \widetilde{P}^n_{X|Z}$ such that the Radon-Nikodym derivative $f \equiv \frac{dP^n_{X|Z}}{d\widetilde{P}^n_{X|Z}}$ is well-defined. Notice that $\widetilde{P}^n_{X|Z} = \widetilde{P}^n_{X|Z=z_1} \times \cdots \times \widetilde{P}^n_{X|Z=z_n}$ and $P^n_{X|Z} = P^n_{X|Z=z_1} \times \cdots \times P^n_{X|Z=z_n}$ such that

$$\mathcal{D}_{KL}\left(P^n_{X|Z}\middle\|\widetilde{P}^n_{X|Z}\right) = \sum_{i=1}^{n} \mathcal{D}_{KL}\left(P^n_{X|Z=z_i}\middle\|\widetilde{P}^n_{X|Z=z_i}\right). \tag{8.5}$$

It is therefore sufficient to show that $\mathcal{D}_{KL}\left(P^n_{X|Z=z_i}\middle\|\widetilde{P}^n_{X|Z=z_i}\right)$ diminishes for one point z_i for increasing sample sizes.

Therefore, for X in $\mathcal{X}$ sampled according to $P^n_{X|Z=z_i}$ or $\widetilde{P}^n_{X|Z=z_i}$, respectively, for a $r \geq 0$, we define

$$\begin{aligned} P^n_{X|Z=z_i}(x, z_i, r) &= P^n_{X|Z=z_i}\left(\{x \in \mathcal{X} : \|(x, z_i) - (x, z_i)\|_\infty \leq r\right), \text{ or} \\ \widetilde{P}^n_{X|Z=z_i}(x, z_i, r) &= \widetilde{P}^n_{X|Z=z_i}\left(\{x \in \mathcal{X} : \|(x, z_i) - (x, z_i)\|_\infty \leq r\right), \end{aligned} \tag{8.6}$$

respectively, which is possible due to *(A2)*.

Then, we partition $\mathcal{X} \times \mathcal{Z}$ into three disjoint sets:

1) $\Omega_1 = \{(x, z_i) \in \mathcal{X} \times \mathcal{Z} : f = 0\}$;
2) $\Omega_2 = \{(x, z_i) \in \mathcal{X} \times \mathcal{Z} : f > 0, P^n_{X|Z=z_i}(x, z_i, 0) > 0\}$;
3) $\Omega_3 = \{(x, z_i) \in \mathcal{X} \times \mathcal{Z} : f > 0, P^n_{X|Z=z_i}(x, z_i, 0) = 0\}$;

such that $\mathcal{X} \times \mathcal{Z} = \Omega_1 \cup \Omega_2 \cup \Omega_3$.

Using the law of total expectation and properties of integrals, we have

$$\begin{aligned} \mathcal{D}_{KL}\left(P^n_{X|Z=z_i}\middle\|\widetilde{P}^n_{X|Z=z_i}\right) &= \int \log(f(x, z_i))\; dP^n_{X|Z=z_i}(x, z_i) && (8.7) \\ &= \int_{\Omega_1} \log(f(x, z_i))\; dP^n_{X|Z=z_i}(x, z_i) && (8.8) \\ &\quad + \int_{\Omega_2} \log(f(x, z_i))\; dP^n_{X|Z=z_i}(x, z_i) && (8.9) \\ &\quad + \int_{\Omega_3} \log(f(x, z_i))\; dP^n_{X|Z=z_i}(x, z_i). && (8.10) \end{aligned}$$

Next, we consider each Ω_1, Ω_2, and Ω_3 in three cases, respectively.

Case 1: Let $(x, z_i) \in \Omega_1$ and $\omega_X(\Omega_1) = \{(x) : (x, z_i) \in \Omega_1\}$ be the projection onto the the first coordinate of Ω_1. Using the definition of f as the Radon-Nikodym derivative, we have

$$P^n_{X|Z=z_i}(\omega_X(\Omega_1)) = \int_{\omega_X(\Omega_1)} f \, dP^n_{X|Z=z_i} = \int_{\omega_X(\Omega_1)} 0 \, dP^n_{X|Z=z_i} = 0,$$

so $\int_{\Omega_1} \log(f(x, z_i)) \, dP^n_{X|Z=z_i}(x, z_i) = 0$, see (8.8).

Case 2: Let $(x, z_i) \in \Omega_2$, i.e., we consider the partition of discrete points as the singletons have a positive measure in $\mathcal{X} \times \mathcal{Z}$. In this context, analogously to [117, Lem. 8], we have

$$f(x, z_i) = \frac{P^n_{X|Z=z_i}(x, z_i, 0)}{\widetilde{P}^n_{X|Z=z_i}(x, z_i, 0)}.$$

Hence, it remains to show that, for $\frac{k_{perm,n}}{n} \to 0$ as $n \to \infty$, $P^n_{X|Z=z_i}(x, z_i, 0) \equiv \widetilde{P}^n_{X|Z=z_i}(x, z_i, 0)$. Let σ_i be the distance from z_i to its k_{perm}-nearest neighbors, see Algorithm 2, line 3. We proceed in the two cases $\sigma_i > 0$ and $\sigma_i = 0$ for $i = 1, \ldots, n$.

First, for $\sigma_i > 0$, i.e., there are less than k_{perm} points in the sample equal to z_i, we show that $\mathbb{P}(\sigma_i > 0) \to 0$, as $n \to \infty$. In particular, the number of points exactly equal to z_i has a binomial distribution with parameters, $n-1$ and $P_Z(z_i)$, $Binomial(n-1, P_Z(z_i))$. Because $\frac{k_{perm,n}}{n} \to 0$ as $n \to \infty$ *(A4)*, there must be n sufficiently large such that $\frac{k_{perm,n}-1}{n-1} \leq P_Z(z_i)$. Therefore, $\mathbb{P}(\sigma_i > 0) = \mathbb{P}(Binomial(n-1, P_Z(z_i)) \leq k_{perm,n} - 1) \to 0$ as $n \to \infty$.

Second, for $\sigma_i = 0$, there must be k_{perm} or more points exactly equal to z_i. In this context, $|\tilde{\mathbf{z}}_i|$ is the total number of points equal to z_i, see Algorithm 2, line 4. Then, we draw samples according to $\widetilde{P}^n_{X|Z=z_i}$ by locally permuting only the $|\tilde{\mathbf{z}}_i|$ samples of x in (x, z) for which $z_j = z_i$, $j \neq i$, i.e., $(x_{\pi^i_m(i)})^n_{i=1}$ where π^i_m is the permutation of indices $\{j : \|(z_j) - (z_i)\|_\infty = 0, j \neq i\}$. Therefore, for all $j \in \tilde{\mathbf{z}}_i$, it holds that $\|(x_{\pi^i_m(j)}, z_i) - (x_i, z_i)\|_\infty = \|(x_j, z_i) - (x_i, z_i)\|_\infty$, see (8.6), i.e., $P^n_{X|Z=z_i}(x, z_i, 0) \equiv \widetilde{P}^n_{X|Z=z_i}(x, z_i, 0)$. Hence, the local CP scheme locally preserves the distribution of X such that, for $\frac{k_{perm,n}}{n} \to 0$ as $n \to \infty$, $P^n_{X|Z=z_i}(x, z_i, 0) \equiv \widetilde{P}^n_{X|Z=z_i}(x, z_i, 0)$ locally for $Z = z_i$. Using basic probability rules it follows that, for $n \to \infty$, $f = 1$ almost surely such that $\int_{\Omega_2} \log(f(x, z_i)) \, dP^n_{X|Z=z_i}(x, z_i) \to 0$ as $n \to \infty$, see (8.9).

Case 3: Let $(x, z_i) \in \Omega_3$, i.e., we consider the continuous partition because the singletons have a zero measure in $\mathcal{X} \times \mathcal{Z}$. In this context, for $n \to \infty$, there are enough samples such that $P^n_{XZ}(\{(x, z_i) \in \Omega_3 : |\tilde{\mathbf{z}}_i| \to k_{perm}\}) = 1$, cf. [117, Lem. 5]. As *(A1)* and *(A2)* holds, analogously to [117, Lem. 7], we have that, for all $\epsilon > 0$,

$$\lim_{r \to 0} \mathbb{P}\left(\left| \frac{P^n_{X|Z=z_i}(x, z_i, r)}{\widetilde{P}^n_{X|Z=z_i}(x, z_i, r)} - f(x, z_i) \right| \leq \epsilon \right) = 1.$$

Hence, for all $\epsilon > 0$, there exists an $r_\epsilon > 0$ such that for all $r \leq r_\epsilon$ it holds that $\mathbb{P}\left(\left| \frac{P^n_{X|Z=z_i}(x, z_i, r)}{\widetilde{P}^n_{X|Z=z_i}(x, z_i, r)} - f(x, z_i) \right| \leq \epsilon \right) = 1$.

Note that $\|(x_j, z_j) - (x_i, z_i)\|_\infty \geq \|(z_j) - (z_i)\|_\infty$ for $j \neq i$. Therefore, let σ_i be the distance of z_i to its nearest neighbors such that $\|(x_{\pi^i_m(i)}, z_i) - (x_i, z_i)\|_\infty = r_\epsilon$. Then, we proceed in the two cases $\sigma_i : \|(x_{\pi^i_m(j)}, z_i) - (x_i, z_i)\|_\infty > r_\epsilon$ and $\sigma_i : \|(x_{\pi^i_m(i)}, z_i) - (x_i, z_i)\|_\infty = r_\epsilon$ for $i = 1, \ldots, n$.

First, we consider $\sigma_{r_\epsilon, i} : \|(x_{\pi^i_m(j)}, z_i) - (x_i, z_i)\|_\infty > r_\epsilon$, i.e., that shuffling x within the distance of $\sigma_{r_\epsilon, i}$ in Z yields a distance greater than r_ϵ. Then, we show that $\mathbb{P}(\{\sigma_i : \|(x_{\pi^i_m(j)}, z_i) - (x_i, z_i)\|_\infty > r_\epsilon\}) \to 0$ as $n \to \infty$. This can only happen when $k_{perm} - 1$ or less neighbors fall within the radius of $\sigma_{r_\epsilon,i}$ such that $\sigma_{r_\epsilon,i} > \sigma_{k_{perm},i}$ where $\sigma_{k_{perm},i}$ denotes the distance of z_i to its k_{perm} nearest neighors, see Algorithm 2, line 3. In this case, $|\tilde{\mathbf{z}}_i| < k_{perm}$. As there are $n-1$ i.i.d. points z_j, $j \neq i$, that can potentially fall into this region with probability $P^n_Z(z_i, \sigma_{r_\epsilon,i}) = P^n_Z(\{z_j \in \mathcal{Z} : \|(z_j) - (z_i)\|_\infty \leq \sigma_{r_\epsilon,i}\})$. Hence, this follows a binomial distribution with parameters $n-1$ and $P^n_Z(z_i, \sigma_{r_\epsilon,i})$. Because $\frac{k_{perm,n}}{n} \to 0$ as $n \to \infty$ *(A4)*, there must be n sufficiently large such that $\frac{k_{perm,n}-1}{n-1} \leq P^n_Z(z_i, \sigma_{r_\epsilon,i})$. Therefore, $\mathbb{P}(\sigma_{k_{perm},i} > \sigma_{r_\epsilon,i}) = \mathbb{P}(Binomial(n-1, P^n_Z(z_i, \sigma_{r_\epsilon,i})) \leq k_{perm,n} - 1) \to 0$ as $n \to \infty$.

Second, for $\sigma_{r_\epsilon,i} : \|(x_{\pi^i_m(j)}, z_i) - (x_i, z_i)\|_\infty = r_\epsilon$, we have that $\sigma_{k_{perm},i} \leq \sigma_{r_\epsilon,i}$. Hence, there must be exactly k_{perm}-nearest neighbors j, $j \neq i$, of z_i, i.e., for which $\|(z_j) - (z_i)\|_\infty \leq \sigma_{k_{perm},i}, j \neq i$, holds. In this context, we draw samples according to $\widetilde{P}^n_{X|Z=z_i}$ by locally permuting only the $|\tilde{\mathbf{z}}_i|$ samples of x in (x, z) for which $\{j : \|(z_i) - (z_j)\|_\infty \leq \sigma_{k_{perm},i}, j \neq i\}$. Therefore, for all $j \in \tilde{\mathbf{z}}_i$, it holds that $\|(x_{\pi^i_m(j)}, z_i) - (x_i, z_i)\|_\infty = \|(x_j, z_i) - (x_i, z_i)\|_\infty$, see (8.6), i.e., $P^n_{X|Z=z_i}(x, z_i, 0) \equiv \widetilde{P}^n_{X|Z=z_i}(x, z_i, 0)$. Hence, the local CP scheme locally preserves the distribution of X such that, for $\frac{k_{perm,n}}{n} \to 0$ as $n \to \infty$, $P^n_{X|Z=z_i}(x, z_i, 0) \equiv \widetilde{P}^n_{X|Z=z_i}(x, z_i, 0)$ locally for $Z = z_i$. Therefore, using basic probability rules, we have that, for $n \to \infty$, $f = 1$ almost surely such that $\int_{\Omega_3} \log(f(x, z_i))\, dP^n_{X|Z=z_i}(x, z_i) \to 0$ as $n \to \infty$, see (8.10). □

Note that the second part of the proof shows that the local CP scheme of Algorithm 2 allows to asymptotically estimate $P^n_{X|Z}$, i.e., $P^n_{X|Z} \equiv \widetilde{P}^n_{X|Z}$ for $n \to \infty$.

Next, we show that mCMIkNN has nontrivial power, i.e., can control type II error.

Theorem 2 (Power: Type II Error Control of $\Phi_{perm,n}$).
Let $(x_i, y_i, z_i)^n_{i=1}$ be i.i.d. samples from P_{XYZ}, and assume (A1) - (A4) hold. Then $\Phi_{perm,n}$, with p-value estimated according to Algorithm 2, can control the type II error, i.e., for any desired nominal value $\beta \in \left[\frac{1}{1+M_{perm}}, 1\right]$, when H_1 is true, then

$$\lim_{n\to\infty} \mathbb{E}_{P_{XYZ}}[1 - \Phi_{perm,n}] = 0. \tag{8.11}$$

Hence, note that the power mCMIkNN is naturally bounded according to M_{perm}, i.e., $1 - \beta \leq 1 - {}^{1}/_{(1+M_{perm})}$. The proof follows from the asymptotic consistency of $\hat{I}_n(x; y|z)$ and that the local CP scheme allows for an asymptotic consistent approximation of $P_{X|Z}$.

Proof. Let $(x, y, z) = (x_i, y_i, z_i)^n_{i=1}$ be drawn from P_{XYZ}, let the M_{perm} permutations $(x^{(m)}, y, z) = (x_{\pi_m(i)}, y_i, z_i)^n_{i=1}$, $m = 1, \ldots, M_{perm}$ be drawn from $\widetilde{P}_{XYZ}$ according to the local CP scheme of Algorithm 2. Then, under $H_1 : X \not\perp\!\!\!\perp Y\ |Z$, let $I(X; Y|Z) = c > 0$, such that the consistency of $\hat{I}_n(x; y|z)$ of Corollary 3 guarantees that, for all $\epsilon > 0$, $\lim_{n\to\infty} \mathbb{P}_{P_{XYZ}}\left(\left|\hat{I}_n(x; y|z) - c\right| > \epsilon\right) = 0$. Similarly, for all $\epsilon > 0$ and $m = 1, \ldots, M_{perm}$, $\lim_{n\to\infty} \mathbb{P}_{P_{XYZ}}\left(\left|\hat{I}_n(x^{(m)}; y|z)\right| > \epsilon\right) = 0$ as $I(X^{(m)}; Y|Z) = 0$ by construction of $\widetilde{P}_{XYZ}$ as $k_{perm} = k_{perm,n} \to \infty$ for $n \to \infty$ *(A4)* and as $P_{X|Z} \equiv \widetilde{P}_{X|Z}$ for $\frac{k_{perm,n}}{n} \to 0$.
Therefore, $(\hat{I}_n(x; y|z), I(X^{(m)}; Y|Z)) \xrightarrow{P} (c, 0)$, such that the continuous mapping theorem with $\phi(x, y) = |x-y|$ implies $|\hat{I}_n(x; y|z) - I(X^{(m)}; Y|Z)| \xrightarrow{P} c$, for $I(X; Y|Z) = c > 0$.

Now, we define

$$A_\beta := \left\{(x, y, z), (x^{(1)}, y, z), ..., (x^{(M_{perm})}, y, z) : \frac{1 + \sum_{m=1}^{M_{perm}} \mathbb{I}\{\hat{I}^{(m)}_n \geq \hat{I}_n\}}{1 + M_{perm}} \leq \beta\right\},$$

where $\hat{I}_n = \hat{I}_n(x;y|z)$ and $\hat{I}_n^{(m)} = \hat{I}_n(x^{(m)};y|z)$ i.e., the set where $\Phi_{perm,n} = 1$. Then, by definition of A_β, we have that

$$\mathbb{E}_{P_{XYZ}}[1 - \Phi_{perm,n}] = 1 - \mathbb{P}_{P_{XYZ}}\left((x,y,z),(x^{(1)},y,z),...,(x^{(M_{perm})},y,z) \in A_\beta\right).$$

As $|\hat{I}_n(x;y|z) - I(x^{(m)};y|y)| \xrightarrow{P} c$ for $(x,y,z),(x^{(1)},y,z),...,(x^{(M_{perm})},y,z) \in A_\beta$,

$$\lim_{n\to\infty} \mathbb{P}\left(\left\{\frac{1 + \sum_{m=1}^{M_{perm}} \mathbb{I}\{\hat{I}_n^{(m)} \geq \hat{I}_n\}}{1 + M_{perm}} \leq \beta\right\}\right) = \mathbb{P}\left(\left\{\frac{1}{1 + M_{perm}} \leq \beta\right\}\right).$$

This completes the proof, as we can conclude that $\lim_{n\to\infty} \mathbb{E}_{P_{XYZ}}[1 - \Phi_{perm,n}] = 1 - 1 = 0$ for all $\beta \in \left[\frac{1}{1+M_{perm}}, 1\right]$. □

Hence, the power $1-\beta$ of `mCMIkNN` is, as common for permutation tests, naturally bounded according to the number of permutations M_{perm}, i.e., $1 - \beta \leq 1 - \frac{1}{1+M_{perm}}$. Therefore, increasing M_{perm} yields more power but comes along with a longer runtime.

Note, although Theorem 2 shows that `mCMIkNN` is asymptotically able to control type II errors, the dimensionality-biasedness of $\hat{I}_n(x;y,|z)$ for $d_Z \to \infty$ affects the robustness in finite sample sizes. In particular, for finite n, $\hat{I}_n(x;y,|z)$ converges in probability towards zero for $d_Z \to \infty$, hence, increasing type II errors, see A_β. In this context, the extensive synthetic evaluation in Chapter 9 indicates that `mCMIkNN` is robust regarding type II errors in the finite case, too.

Therefore, our work is in line with the result of Shah and Peters [162] and Kim et al. [78] by demonstrating that, under the mild assumptions `(A1)` and `(A2)` which allow approximating $P_{X|Z}$, one can derive a CI test that is valid (see Theorem 1), and has nontrivial power (see Theorem 2).

8.4.4 `mCMIkNN`-Based Constraint-Based Causal Discovery

We examine the asymptotic consistency of `mCMIkNN`-based causal discovery, in particular, using the well-known PC algorithm [168]. Note that constraint-based methods for causal discovery cannot distinguish between different directed acyclic graphs (DAGs) $\mathcal{G}$ in the same equivalence class (see Section 1.2 on page 3). Hence, the PC algorithm aims to find thecomplete partially directed acyclic graph (CPDAG), denoted with $\mathcal{G}_{CPDAG}$, that represents the Markov equivalence class of the true DAG $\mathcal{G}$. Constraint-based methods apply CI tests to test whether $X \perp\!\!\!\perp Y \mid Z$ for $X, Y \in \mathbf{V}$ with $d_X = d_Y = 1$, and $Z \subseteq \mathbf{V} \setminus \{X, Y\}$ iteratively with increasing d_Z given a nominal value α to estimate the undirected skeleton of $\mathcal{G}$ and corresponding separation sets in the first step. In a second step, orienting as many of the undirected edges through the repeated application of deterministic orientation rules yields $\hat{\mathcal{G}}_{CPDAG}(\alpha)$ [168, 74].

Using `mCMIkNN` for constraint-based causal discovery allows consistently estimating the $\mathcal{G}_{CPDAG}$ for $n \to \infty$.

Theorem 3 (Consistency of `mCMIkNN`-Based Causal Discovery).
Let $\mathbf{V}$ be a finite set of variables with joint distribution $P_\mathbf{V}$ and assume (A1) - (A4) hold for all $X, Y \in \mathbf{V}$ and $Z \subseteq \mathbf{V} \setminus \{X, Y\}$. Further, assume the general assumptions of the PC algorithm hold, i.e., sufficiency, causal faithfulness, and causal Markov condition, see [168]. Let $\hat{\mathcal{G}}_{CPDAG,n}(\alpha_n)$ be the estimated CPDAG of the PC algorithm and $\mathcal{G}_{CPDAG}$ the CPDAG of the true underlying DAG $\mathcal{G}$. Then, for $\alpha_n = \frac{1}{1+M_{perm,n}}$ with $M_{perm,n} \to \infty$ as $n \to \infty$,

$$\lim_{n\to\infty} \mathbb{P}_{P_\mathbf{V}}\left(\hat{\mathcal{G}}_{CPDAG,n}(\alpha_n) = \mathcal{G}_{CPDAG}\right) = 1. \tag{8.12}$$

The idea of the proof is to consider wrongly detected edges due to incorrect CI decisions and show that they can be controlled asymptotically via an appropriate α_n.

Proof. The idea of the proof is inspired by Kalisch et al. [74] and considers wrongly detected edges due to incorrect CI decisions of `mCMIkNN`. In contrast, we show that the errors due to incorrect CI decisions can be controlled asymptotically by choosing $\alpha_n = \frac{1}{1+M_{perm,n}}$.

In the adjacency search, the first part of the PC algorithm, an error occurs if, for nodes X and Y and conditioning set Z, an error event $E_{X,Y|Z}$ occurs. Thus,

$$\begin{aligned}\mathbb{P}(\text{error occurs in the first part of PC}) &\leq \mathbb{P}\left(\bigcup_{X,Y,Z} E_{X,Y|Z}\right)\\ &\leq \sum_{X,Y,Z} \mathbb{P}(E_{X,Y|Z} \text{ occurs})\\ &\leq N^{N-2} \sup_{X,Y,Z} \mathbb{P}(E_{X,Y|Z} \text{ occurs}),\end{aligned}$$

as the number of combinations of X, Y, and Z in $\mathbf{V}$ is bounded by N^{N-2}.

Now, we split error events into type I and II errors, i.e., $E_{X,Y|Z} = E^{I}_{X,Y|Z} \cup E^{II}_{X,Y|Z}$,

$$\begin{aligned}\text{type I error } E^{I}_{X,Y|Z}: &\quad p_{perm,n} \leq \alpha_n, \text{ and } X \perp\!\!\!\perp Y \mid Z;\\ \text{type II error } E^{II}_{X,Y|Z}: &\quad p_{perm,n} > \alpha_n, \text{ and } X \not\perp\!\!\!\perp Y \mid Z.\end{aligned}$$

Then, the statistical validity of `mCMIkNN` according to Theorem 1 ensures that, for any $\alpha_n \in [0,1]$, we have that $\mathbb{P}(E^{I}_{X,Y|Z} \text{ occurs}) \leq \alpha_n$ for $n \to \infty$. Further, the power of `mCMIkNN` according to Theorem 2 ensures, that for any $\alpha_n \in [\frac{1}{1+M_{perm}}, 1]$, $\mathbb{P}(E^{II}_{X,Y|Z} \text{ occurs}) = 0$ for $n \to \infty$.

Hence, choosing $\alpha_n = \frac{1}{1+M_{perm}}$ with $M_{perm} = M_{perm,n} \to \infty$ as $n \to \infty$, we have that

$$\begin{aligned}\mathbb{P}(\text{error occurs in the first part of PC}) &\leq N^{N-2} \sup_{X,Y,Z} P(E_{X,Y|Z} \text{ occurs})\\ &\leq N^{N-2} \sup_{X,Y,Z} \Big(P(E^{I}_{X,Y|Z} \text{ occurs}) + P(E^{I}_{X,Y|Z} \text{ occurs})\Big)\\ &\leq N^{N-2} \frac{1}{1+M_{perm,n}}\\ &= 0, \text{ as } n \to \infty.\end{aligned}$$

Therefore, the undirected skeleton of $\mathcal{G}$ and separation sets are correctly estimated for $n \to \infty$, which proves Theorem 3, as the edge orientation (second part) of the PC algorithm will never fail, see [115]. □

Note that, as the upper bound on the errors is general for constraint-based methods, the consistency statement of Theorem 3 holds for modified versions of the PC algorithm, e.g., its order-independent version PC-stable [20], too. Hence, using `mCMIkNN` for constraint-based causal discovery allows consistently estimating the CPDAG $\mathcal{G}_{CPDAG}$ of the true underlying DAG $\mathcal{G}_{CPDAG}$ for $n \to \infty$. In particular, we showed consistency under mild assumptions on $P_{\mathbf{V}}$, cf. Kalisch and Bühlmann [74] requiring multivariate Gaussianity.

8.4.5 mCMIkNN: Assumptions and Computational Complexity

In this section, we provide more information on the assumptions, the computational complexity, and its implications for application.

First, recap all assumptions on P_{XYZ} and parameters k_{CMI} and k_{perm} of `mCMIkNN`.

Assumptions 1. *Let $(\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}, \mathcal{B}, P_{XYZ})$ be a probability space defined on the metric space $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ with dimensionality $d_x + d_y + d_z$, equipped with the Borel σ-algebra $\mathcal{B}$, and a regular joint probability measure P_{XYZ}. Throughout this work, we assume:*

- *(A1) $P_{XY|Z}$ is non-singular such that $f \equiv \frac{dP_{XY|Z}}{d(P_{X|Z} \times P_{Y|Z})}$ is well-defined, and assume, for some $C > 0$, $f(x, y, z) < C$ for all $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$;*
- *(A2) $\{(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} : P_{XYZ}((x, y, z)) > 0\}$ countable and nowhere dense in $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$;*
- *(A3) $k_{CMI} = k_{CMI,n} \to \infty$ and $\frac{k_{CMI,n}}{n} \to 0$ as $n \to \infty$;*
- *(A4) $k_{perm} = k_{perm,n} \to \infty$ and $\frac{k_{perm,n}}{n} \to 0$ as $n \to \infty$.*

In the following, we examine the above assumptions in more detail.

(A1): While rather technical, non-singularity is helpful for practice as it provides a sufficient condition that can be verified in data analysis.

Definition 1 (Non-Singularity of $P_{XY|Z}$).
Let $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}, \mathcal{B}, P_{XYZ}$ be a probability space with marginal conditional probability measures, $P_{X|Z}$ and $P_{Y|Z}$. $P_{XY|Z}$ is non-singular if for any measurable set, $E \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, $a \in X \times Z$ and $b \in Y \times Z$, such that $P_{X|Z}(E_b) = 0$ and $P_{Y|Z}(E_a) = 0$, then $P_{XY|Z}(E) = 0$, where $E_b = \{(x, z) : (x, b, z) \in E\}$ and $E_a = \{(y, z) : (a, y, z) \in E\}$.

Assuming non-singularity of $P_{XY|Z}$ ensures absolute continuity $P_{XY|Z} \ll P_{X|Z} \times P_{Y|Z}$, i.e., the existence of the Radon-Nikodym derivative $f \equiv \frac{dP_{XY|Z}}{d(P_{X|Z} \times P_{Y|Z})}$ using Fubini's theorem and Radon-Nikodym's theorem, see [117, Thm. 2.2]. Further, given that f is well-defined, the existence of a $C > 0$ such that $f(x, y, z) < 0$ for all $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ is satisfied whenever the distribution is (i) (finitely) discrete, (ii) continuous, (iii) some dimensions are (countably) discrete and some are continuous, and (iv) a mixture of the previous cases, see [43]. Hence, for practice, checking the sufficient condition of non-singularity can be done by ensuring that there exists no set $E \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ such that $P_{XY|Z}(E) = 0$ while $P_{Y|Z}(E_a) = 0$ for $E_b = \{(x, z) : (x, b, z) \in E\}$ and $E_a = \{(y, z) : (a, y, z) \in E\}$, see [42, 199].

(A2): Assumption *(A2)* is satisfied whenever the distribution of P_{XYZ} is (i) (finitely) discrete, (ii) continuous, (iii) some dimensions are (countably) discrete and some are continuous, and (iv) a mixture of the previous cases, which covers most real-world data [43]. This mild assumption simplifies the application of `mCMIkNN` in practice. In contrast, stronger assumptions such as lower bounds on corresponding probabilities for discrete points, see [3, 78], or further smoothness assumptions for continuous variables, see [198, 7], allow examining tighter bounds on type I and II error control for the finite case.

(A3): The kNN-parameter k_{CMI} can be seen as the lower bound of a locally data-adaptive "bandwidth" parameter used in the local kNN-based estimation of the Shannon entropies, see [148]. In contrast to global bandwidths of kernel-based measures, which require a careful adjustment, particularly in mixed discrete-continuous data, k_{CMI} is

locally adapted within the density estimation for each sample point (see Algorithm 1 and Algorithm 2) providing easier calibration of the CI test.

In this context, k_{CMI} can be chosen given the data characteristic[8]. In particular, higher ratios of discrete variables require smaller values of k_{CMI}. Overall, k_{CMI} can be increased for increasing n, e.g., using Runge's rule of thumb $k_{CMI} \approx 0.1n, \ldots, 0.2n$, particularly for low discrete variable ratios, see [148]. For more information, see the detailed evaluation results on mCMIkNN's calibration in Section 9.2 on page 105.

Therefore, although *(A3)* requires $k_{CMI} \to \infty$ for $n \to \infty$, needed to receive asymptotic results, small values of k_{CMI} already suffice to approximate the densities well. In particular, the experimental results for $n \in \{50, \ldots, 1\,000\}$ provided in Section 9.2 indicate that fixing the value to $k_{CMI} = 25$ yields well-calibrated tests while not affecting power much for the finite case.

(A4): The kNN-parameter k_{perm} is used to simulate the null distribution as local permutations are drawn within the k_{perm}-nearest distance regarding the neighborhood of $Z = z$. Therefore, k_{perm} should be chosen given the data characteristics similar to k_{perm}. For more information, see the detailed evaluation results on mCMIkNN's calibration in Section 9.2 on page 105.

In this context, too large values of k_{perm} (or even a fully non-local permutation with $k_{perm} \approx n$) destroy the conditional marginal distributions under H_0, hence, increase type I errors, and too small values of k_{perm} are not sufficient to simulate H_0 given H_1 accurately, hence, increase type II errors[8].
In our experimental results for $n \in \{50, \ldots, 1\,000\}$ provided in Section 9.2, we find that small values of $k_{perm} \approx 5$ already suffice to simulate the null distribution reliably, as the local data-adaptiveness yields robustness.

Second, we consider the local CP parameter M_{perm} and examine mCMIkNN's computational complexity in more detail.

M_{perm}: As commonly done for permutation tests, the number of permutations M_{perm} used for the mCMIkNN (see Algorithm 2) is chosen according to the desired nominal value α and respective requirements on the derived p-value $p_{perm,n}$. Further, note that according to Theorem 2, the power $1-\beta$ is naturally bounded by $1 - \frac{1}{1+M_{perm}}$. For example, choosing $M_{perm} = 100$ allows for the smallest possible p-value of approx. 0.099 and bounds the power to be smaller than approx. 0.901. Hence, $M_{perm} = 100$ provides a good starting point given a nominal value $\alpha = 0.05$ and may be increased to receive more power or to examine smaller nominal values. For example, we choose $M_{perm} = 1\,000$ for all experiments with $\alpha = 0.01$, see CI testing in Section 9.2 and Section 9.3, and $M_{perm} = 100$ for all experiments with $\alpha = 0.05$, see causal discovery in Section 9.5. For more information on the choice of M_{perm}, we refer to [38, 137].

Computational Complexity: The main computational cost of mCMIkNN comes from the kNN searches in Algorithm 1 and Algorithm 2, which is $\mathcal{O}(n^2)$ in the worst case. To speed up the searches, mCMIkNN uses k-d-trees, reducing the computational complexity to $\mathcal{O}(n \times log(n))$ when searching over all n samples. For a detailed evaluation of runtimes and a discussion on execution strategies, see Section 9.4 on page 112.

[8] For more details on the impact of k_{CMI} and k_{perm}, we refer to the illustrative examples covering the continuous case provided by Runge [148].

9

Empirical Evaluation

In this chapter, we consider the mixed additive noise model (MANM) (Section 9.1) to synthetically examine `mCMIkNN`'s robustness (Section 9.2). Further, we compare `mCMIkNN`'s empirical performance against state-of-the-art competitors regarding CI decisions (Section 9.3) and their runtimes (Section 9.4), where we sketch parallel execution strategies to speed up `mCMIkNN`, too. Finally, we evaluate the CI tests' accuracy when used in constraint-based causal discovery (Section 9.5).

Contribution: Parts of this chapter have previously been published in the paper [69]. The thesis author developed the evaluation concepts, conducted the experiments, and prepared the original draft. The implementation of the experiments was joint work with Christopher Hagedorn, who also supported the implementation of `mCMIkNN`. The coauthors improved the paper's material and its presentation.

9.1 Synthetic Data Generation using the MANM

In this section, we describe how the mixed additive noise model (MANM) is used to simulate the synthetic data according to conditional independence (CI) characteristics or causal structures (Section 9.1.1) and present the respective parameters of `MANM-CS` used for the sampling process (Section 9.1.2).

9.1.1 Modeling Conditional Independence and Causal Structures

As recommended by Huegle et al. [67] (see Chapter 4 on page 43), we consider the mixed additive noise model (MANM) for evaluating approaches for CI testing and constraint-based causal discovery from mixed discrete-continuous data.

In particular, for all $X \in \mathbf{V}$, let X be generated from its J discrete parents $\mathcal{P}^{dis}(X) \subseteq \mathbf{V} \setminus X$, where $J := \#\mathcal{P}^{dis}(X)$, its K continuous parents $\mathcal{P}^{con}(X) \subseteq \mathbf{V} \setminus X$, where $K := \#\mathcal{P}^{con}(X)$, and an independent noise term N_X according to

$$X = \frac{1}{J} \sum_{j=1,\dots,J} f_j(Z_j) + \left(\sum_{k=1,\dots,K} f_k(Z_k) \right) \bmod d_X + N_X. \tag{9.1}$$

We restrict our attention to the cyclic model, where the domain of a discrete variable is the modulo ring $\mathbb{Z}/d_X\mathbb{Z}$ to restrict the support of discrete variables, i.e., X can take values from $\{0, \dots, d_X - 1\}$. Moreover, the independent noise variable N_X either is a continuous distributed random variable, i.e., $N_X \sim \mathcal{N}(0, 1)$, or discrete distributed over $\mathbb{Z}/d_X\mathbb{Z}$ with $\mathbb{P}(N_X = 0) \geq \mathbb{P}(N_X = l)$ for all $l \in \{0, \dots, d_X - 1\}$ if X is continuous or discrete, respectively. Therefore, $f_j : \mathbb{Z}/d_j\mathbb{Z} \to \mathbb{Z}/d_X\mathbb{Z}$ and $f_k : \mathbb{R} \to \mathbb{Z}/d_X\mathbb{Z}$ if X is discrete, or $f_j : \mathbb{Z}/d_j\mathbb{Z} \to \mathbb{R}$ and $f_k : \mathbb{R} \to \mathbb{R}$ if X is continuous. In particular, functions $f_j : \mathbb{R} \to \mathbb{Z}/d_X\mathbb{Z}$ assign a probability to each realization within the support $\{0, \dots, d_X - 1\}$ of X using a softmax function while $f_k, : \mathbb{R} \to \mathbb{R}$ is a continuous function.

Note that we scale the parents' signals, see Eq. (9.1), to reduce the noise for subsequent variables which avoids high varsortability [143], and max-min normalize all continuous variables of the dataset.

Hence, by construction (A1) and (A2) hold true for all combinations of $X, Y \in \mathbf{V}$ and $Z \subseteq \mathbf{V} \setminus \{X, Y\}$. For more information on MANMs, see Chapter 4 (page 43).

9.1.2 Parameters of `MANM-CS`

In Table 9.1, we describe the parameters and their values used to generate synthetic data with the `MANM-CS` library.

For the first four experiments, i.e., calibration (Section 9.2.1), robustness (Section 9.2.2), CI testing (Section 9.3), and runtime comparison (Section 9.4) a directed acyclic graph (DAG) is generated according to $X \perp\!\!\!\perp Y |\, Z$ or $X \not\perp\!\!\!\perp Y | Z$ and the mixed additive noise model (MANM) is sampled according to the parameters at the top. Hence, we generate CGMs that either directly induce CI characteristics between variables X and Y conditioned on $Z = \{Z_1, \dots, Z_{d_Z}\}$, d_Z between 1 and 7 (see Section 9.2 - 9.4). Moreover, we consider different ratios of discrete variables between 0 and 1. We restrict our attention to the cyclic model with d_X, d_Y, and $d_{Z_i} \in \{2, 3, 4\}$ for discrete X, Y, and Z_i, $i = 1, \dots, d_Z$, and continuous functions that are equally drawn from $\{id(\cdot), (\cdot)^2, cos(\cdot)\}$. Hence, the n observational samples drawn from the respective CGMs allow for a comprehensive empirical evaluation of the CI tests' accuracy.

For the experimental evaluation of causal discovery (Section 9.5), the parameters at the bottom are used to generate the structure of the DAGs, too.

Table 9.1: Parameters of `MANM-CS` used for synthetic data generation. For CI test experiments, we use appropriate DAGs that directly induce $X \perp\!\!\!\perp Y |\, Z$ or $X \not\perp\!\!\!\perp Y | Z$ and generate a CGM according to the parameters at the top. For the causal discovery, the parameters at the bottom are used to generate the structure of the DAGs, too.

Parameter Description	Values
ratio of discrete variables	$\{0.0, 0.25, 0.5, 0.75, 1.0\}$
range for discrete classes	$\{2, 3, 4\}$
discrete signal to noise ratio	$\{0.85\}$
continuous functions with sample probabilities	$\{(\frac{1}{3}, id(\cdot)), (\frac{1}{3}, (\cdot)^2), (\frac{1}{3}, \cos(\cdot))\}$
standard deviation of continuous Gaussian noise	$\{1.0\}$
scale parents	$\{1\}$
number of samples	$\{50, 100, 250, 500, 1\,000\}$
variables scaling	{`normal`}
number of variables N	$\{10, 20, 30\}$
edge density of the CGMs	$\{0.1, 0.2, 0.3, 0.4\}$

9.2 Calibration and Robustness of `mCMIkNN`

In this section, we provide recommendations for calibrating `mCMIkNN` (Section 9.2.1) and show its robustness, i.e., the ability to control type I and II errors in the finite case (Section 9.2.2).

9.2.1 Calibration of `mCMIkNN`

To receive a recommendation for calibrating `mCMIkNN`, we evaluate the accuracy of CI decisions for different combinations of k_{CMI} and k_{perm}.

Therefore, we restrict our attention to two simple DAGs $\mathcal{G}$ with variables $\mathbf{V} = \{X, Y, Z_1, ..., Z_{d_Z}\}$, where first, X and Y have common parents $Z = \{Z_1, ..., Z_{d_Z}\}$ in $\mathcal{G}$, i.e., $H_0 : X \perp\!\!\!\perp Y|\boldsymbol{Z}$, and second, there exists an additional edge connecting X and Y in $\mathcal{G}$, i.e., $H_1 : X \not\perp\!\!\!\perp Y|Z$, see Section 9.1.1. Accordingly, we generate the data using the MANM model with parameters described in Section 9.1.2.

To get a balanced view on the accuracy of `mCMIkNN`'s CI decisions, we compare the area under the receiver operating curve (ROC AUC) given varying parameters k_{CMI} and k_{perm}. In Table 9.2, we present a detailed comparison of the ROC AUCs for different combinations of $k_{CMI} \in \{5, 25, 100, 200\}$ and $k_{perm} \in \{5, 25, 100, 200\}$ with sample sizes ranging from 50 to 1 000. Note that we consider $\alpha = 0.05$ and set $M_{perm} = 100$, which provides a good starting point (see Section 8.4.5). Further, Table 9.3 (on page 106) presents the type I and type II errors for the same set of CI decisions used in Table 9.2.

Table 9.2: ROC AUC scores (higher better) for different combinations of k_{CMI}, k_{perm}, and samples n with fixed $M_{perm}=100$ and $\alpha = 0.05$ derived from CI decisions over multiple settings, e.g., sampled with a varying dimension of Z, $d_Z \in \{1, 3, 5, 7\}$, continuous functions, or discrete variable ratios (for the corresponding parameters of `MANM-CS`, see Table 9.1).

ROC AUC Scores					
samples n	k_{perm} / k_{CMI}	5	25	100	200
50	5	**0.58**	**0.58**	-	-
	25	0.56	0.55	-	-
100	5	**0.64**	**0.64**	-	-
	25	**0.64**	**0.64**	-	-
250	5	0.72	0.72	0.72	0.72
	25	**0.73**	**0.73**	**0.73**	**0.73**
	100	0.66	0.65	0.64	0.64
	200	0.55	0.54	0.53	0.53
500	5	**0.77**	**0.77**	**0.77**	**0.77**
	25	**0.77**	0.76	0.76	0.76
	100	0.73	0.71	0.71	0.7
	200	0.67	0.66	0.65	0.65
1 000	5	0.8	0.8	0.8	0.8
	25	**0.81**	0.8	0.8	0.8
	100	0.77	0.75	0.74	0.74
	200	0.73	0.71	0.7	0.69

Table 9.3: Type I (top) and type II (bottom) error rates (smaller better) for different combinations of k_{CMI}, k_{perm}, and samples n with fixed $M_{perm} = 100$ derived from CI decisions ($\alpha = 0.05$) over multiple settings, e.g., sampled with a varying dimension of Z, $d_Z \in \{1, 3, 5, 7\}$, continuous functions, or discrete variable ratios (for the corresponding parameters of `MANM-CS`, see Table 9.1).

Type I Error Rates					
samples n	k_{perm} k_{CMI}	5	25	100	200
50	5	0.06	0.06	-	-
	25	**0.04**	0.06	-	-
100	5	**0.05**	0.06	-	-
	25	**0.05**	0.06	-	-
250	5	0.07	0.07	0.07	0.07
	25	0.07	0.08	0.09	0.09
	100	0.06	0.08	0.09	0.09
	200	**0.04**	0.07	0.09	0.09
500	5	**0.07**	**0.07**	**0.07**	**0.07**
	25	0.08	0.11	0.11	0.12
	100	0.08	0.11	0.12	0.13
	200	**0.07**	0.1	0.12	0.11
1 000	5	**0.08**	**0.08**	0.09	**0.08**
	25	0.12	0.15	0.15	0.15
	100	0.12	0.17	0.18	0.19
	200	0.1	0.15	0.18	0.18

Type II Error Rates					
samples n	k_{perm} k_{CMI}	5	25	100	200
50	5	**0.78**	**0.78**	-	-
	25	0.84	0.84	-	-
100	5	**0.66**	**0.66**	-	-
	25	0.67	**0.66**	-	-
250	5	0.49	0.49	0.5	0.49
	25	0.47	**0.46**	**0.46**	**0.46**
	100	0.63	0.62	0.62	0.62
	200	0.86	0.85	0.85	0.85
500	5	0.39	0.38	0.38	0.38
	25	**0.37**	**0.37**	**0.37**	**0.37**
	100	0.47	0.46	0.46	0.46
	200	0.58	0.58	0.58	0.58
1 000	5	0.31	0.31	0.31	0.31
	25	**0.26**	**0.26**	**0.26**	**0.26**
	100	0.34	0.34	0.34	0.34
	200	0.43	0.43	0.44	0.43

For k_{CMI}, the detailed evaluation shows that small values of k_{CMI}, e.g., $k_{CMI} \leq 25$, are sufficient to estimate the true CMI value achieving appropriate accuracy (see Table 9.3), i.e., yield higher ROC AUCs for all sample sizes n and k_{perm} (see Table 9.2). This is in line with the results of [117]. Note that for a significantly large number of samples, a common rule-of-thump of $k_{CMI} \approx 0.1n, \ldots, 0.2n$, e.g., see [148]. As the runtime of mCMIkNN increases loglinear with k_{CMI}, fixing k_{CMI} to small values keeps the experimental evaluation practical without affecting the power much.

For k_{perm}, the ROC AUCs marginally decrease with larger values of k_{perm} (see Table 9.2), such that small values already suffice to simulate the null distribution reliably (see Table 9.3), which is in line with results from [148]. Hence, our experimental evaluation indicates that the power of mCMIkNN is relatively robust regarding the choice of k_{perm} (of course, as long as $k_{perm} < n$). In this context, note that the runtime is only marginally affected by k_{perm}.

Hence, the experimental results indicate that fixing the values to $k_{CMI} = 25$ and $k_{perm} = 5$ yields well-calibrated CI tests while not affecting accuracy much for the finite case. For more information on the parameters k_{CMI} and k_{perm}, see Section 8.4.5 on page 100 or see the illustrative examples covering the continuous case provided by Runge [148].

9.2.2 Robustness of mCMIkNN: Type I and II Error Control

We evaluate mCMIkNN's robustness regarding validity and power in the finite case by examining the type I and II error rates as depicted in Fig. 9.1 (on page 108). Therefore, we again restrict our attention to the two simple CGMs $\mathcal{G}$ with variables $\mathbf{V} = \{X, Y, Z_1, ..., Z_{d_Z}\}$, where first, $H_0 : X \perp\!\!\!\perp Y|Z$, and second, $H_1 : X \not\perp\!\!\!\perp Y|Z$ for $Z = \{Z_1, ..., Z_{d_Z}\}$ holds true.

We see that mCMIkNN can control type I errors for all discrete variable ratios dvr (Fig. 9.1 vertical) and sizes of the conditioning set d_Z (Fig. 9.1 horizontal). Moreover, for an increasing number of samples n, the type II error rates decrease (Fig. 9.1 in each subplot), hence, mCMIkNN achieves non-trivial power, particularly for small sizes of the conditioning sets d_Z. In this context, higher type II errors in the case of higher dimensions d_Z point out that mCMIkNN suffers from the curse of dimensionality, see the dimensionality-biasedness of $\hat{I}_n(X;Y|Z)$ for increasing d_Z as shown in Corollary 3 on page 91.

In summary, the empirical results are in line with the theoretical results on the asymptotic type I and II error control, see Theorem 1 on page 93 and Theorem 2 on page 97.

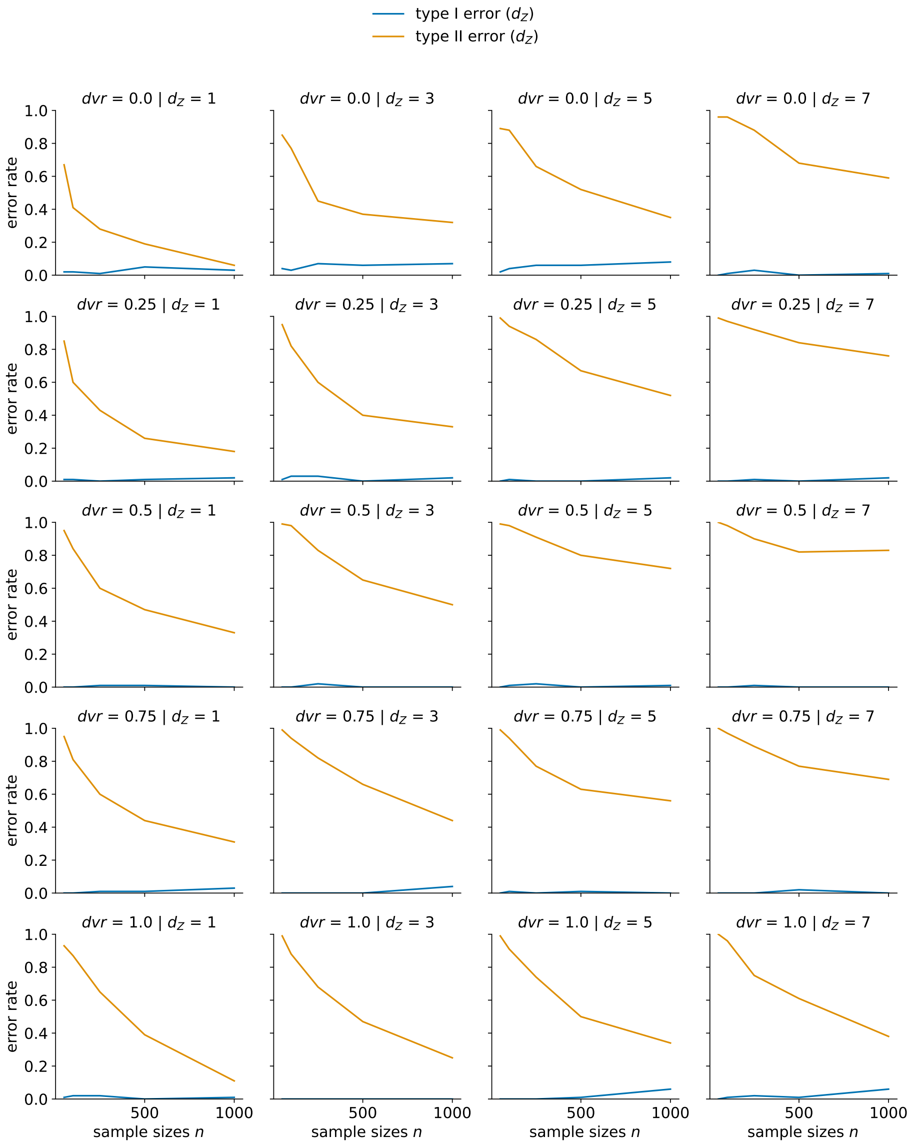


Fig. 9.1: Type I and II error rates of `mCMIkNN` (smaller better) given varying sample sizes n, each subplot illustrates one combination of a dimension of Z, i.e., $d_Z \in \{1, 3, 5, 7\}$ (left to right), and a distinct discrete variable ratio $dvr \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$ (top to bottom).

9.3 Conditional Independence Testing

In this section, we examine `mCMIkNN`'s empirical performance compared to state-of-the-art CI tests valid for mixed discrete-continuous data.

In particular, we chose the following CI tests and implementations:

- `mCMIkNN` our kNN-based CI using CMI and the local permutation scheme;
- `CG` a likelihood ratio test assuming conditional Gaussianity [2] implemented as function `mixCItest` in the R package `micd` [39];
- `KCIT` the well-known kernel-based CI test from Zhang et al. [195];

- `disc`χ^2 a discretization-based approach, where we discretize continuous variables using `Ckmeans.1d.dp` from the same-named R package [184] (using BIC to choose an appropriate number of clusters $k \in \{4, ..., 9\}$), before applying Pearson's χ^2 test from the R package `pcalg` [75];
- `aHist`χ^2 a non-parametric CI test, where CMI is estimated based upon adaptive histograms [112] and a CI test is derived via a pseudo-p-value using a χ^2 correction coded for $\alpha = 0.01$ (see `CMIp.Chisq95` [112])[9].

In the following experiments, we again consider the two CGMs used for the calibration in Section 9.2.2 and examine the respective ROC AUC scores (Section 9.3.1) and type I and II errors (Section 9.3.2) of 20 000 CI decisions for $\alpha = 0.01$.

9.3.1 ROC AUC Scores of CI Decisions

In Fig. 9.2, we compare the CI tests' area under the receiver operating curve (ROC AUC) for various sample sizes n (top left), sizes of the conditioning set d_Z (top right), and ratios of discrete variables dvr (bottom).

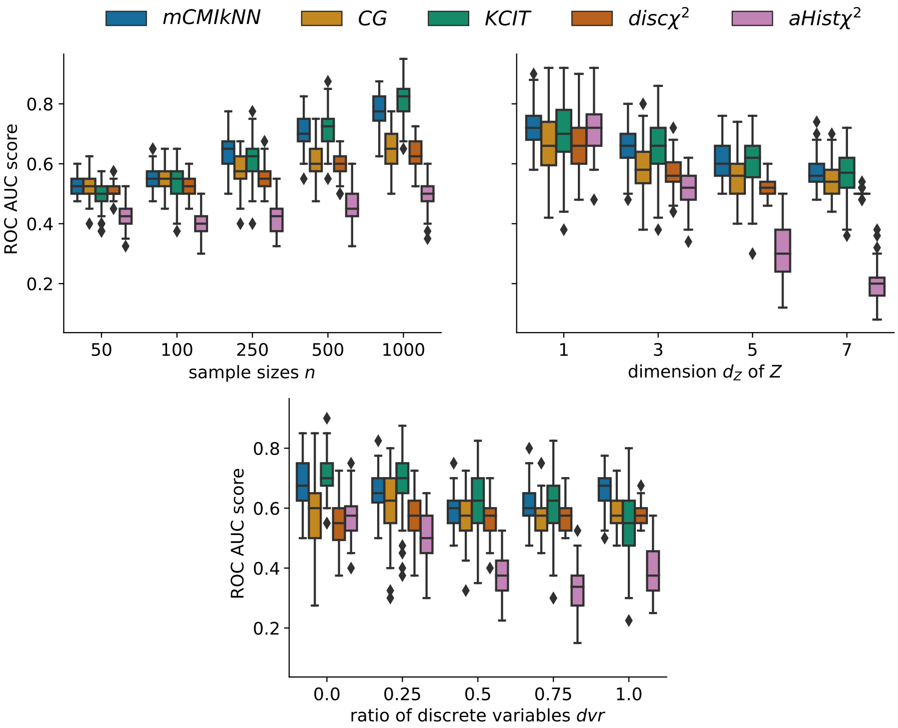


Fig. 9.2: ROC AUC scores (higher better) of 20 000 CI decisions of the CI tests `mCMIkNN`, `CG`, `KCIT`, `disc`χ^2, and `aHist`χ^2 with varying sample sizes n (top left), sizes of discrete domains d_Z (top right), and ratios of discrete variables dvr (bottom)[9].

[9] Note that runs of `aHist`χ^2 are limited to an execution time of 10 minutes and approx. 4 900 out of 20 000 runs for the CI experiment did not complete in time. Further, the implementation does not cover $dvr = 1.0$. Therefore, its long execution time and the restriction to non-discrete data do not allow for usage in constraint-based causal discovery. Hence, `aHist`χ^2 is excluded in the respective experiment.

While the ROC AUC scores of all CI tests increase as n grows (Fig. 9.2 top left), mCMIkNN outperforms all competitors, particularly for small sizes, e.g., $n \leq 500$. With increasing sample sizes, the performance of KCIT catches up to ROC AUC scores of mCMIkNN, e.g., for $n = 1\,000$.

For an increasing size of the conditioning sets d_Z (Fig. 9.2 top right), we observe that all methods suffer from the curse of dimensionality. At the same time, mCMIkNN achieves higher ROC AUC scores than the competitors.

Moreover, mCMIkNN achieves the highest ROC AUC independent of the ratio of discrete variables dvr (Fig. 9.2 bottom), only beaten by KCIT for some dvr's.

Hence, mCMIkNN achieves the overall best ROC AUC scores, particularly for small sample sizes, only beaten by KCIT for high dimensional settings or given primarily continuous distributed variables.

9.3.2 Type I and II Errors of CI Decisions

To receive a more detailed evaluation of the CI tests' statistical validity and power, we directly compare the CI tests' type I and type II error rates.

In particular, we compare the CI tests' type I errors (Fig. 9.3) and type II errors (Fig. 9.4) concerning various sample sizes (top left), different sizes of conditioning sets d_Z (top right), and different ratios of discrete variables dvr (bottom).

To achieve statistical validity, type I error rates should be close to the required nominal value $\alpha = 0.01$, see Theorem 1 on page 93. As depicted in Fig. 9.3, statistical validity

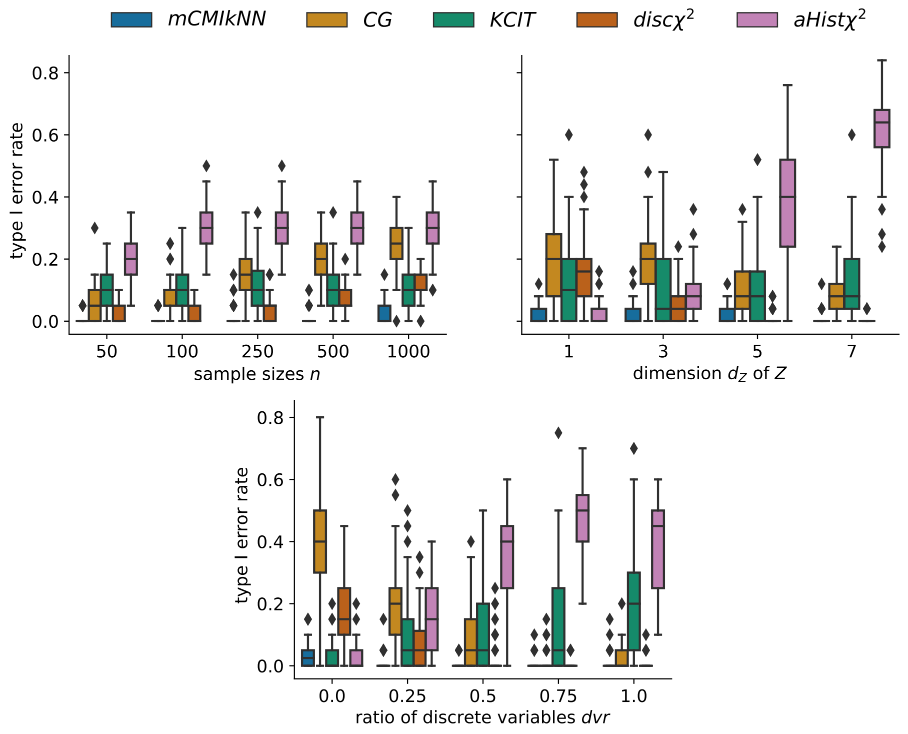


Fig. 9.3: Type I error rate (smaller better) of 20 000 CI decisions of the CI tests mCMIkNN, CG, KCIT, discχ^2, and aHistχ^2 with varying sample sizes n (top left), sizes of discrete domains d_Z (top right), and ratios of discrete variables dvr (bottom)[9].

for all settings can only achieved by mCMIkNN, while all other CI tests struggle with their well-known weaknesses regarding the curse of dimensionality or inadequate assumptions.

For example, aHistχ^2 suffers strongly from the curse of dimensionality (Fig. 9.3 top right), which yields weaknesses in type I error control when examining the aggregated view for increasing number of samples (Fig. 9.3 top left). Further, for a low discrete variable ratio (Fig. 9.3 bottom), CG has high type I error rates as the linearity assumption of the conditional Gaussianity is not fulfilled in the continuous case. Similarly, for low discrete variable ratios (Fig. 9.3 bottom), type I error rates of KCIT are low as kernel-based methods demonstrate their strength in nonlinear continuous data but increase for increasing ratios of discrete variables.

As depicted in Fig. 9.4, the type II error rates align with the ROC AUC scores of Fig. 9.2. In particular, type II error rates of all CI tests decrease as n grows (Fig. 9.4 top left) with the well-known weaknesses regarding the curse of dimensionality (Fig. 9.4 top right) and inadequate assumptions (Fig. 9.4 bottom).

Concerning an increasing size of the conditioning sets d_Z (Fig. 9.4 top right), we observe that all methods suffer from the curse of dimensionality, while KCIT, directly followed by mCMIkNN, can control type II errors for higher dimensions of conditioning sets d_Z. In this context, aHistχ^2 and discχ^2 suffer strongly from the curse of dimensionality as adaptive histogram-based and discretization-based approaches require much more sample sizes to achieve an appropriate power (Fig. 9.4 top).

For varying ratios of discrete variables dvr (Fig. 9.4 bottom), we observe that mCMIkNN and KCIT achieve stable and low type II errors for all dvr. In this context, for the re-

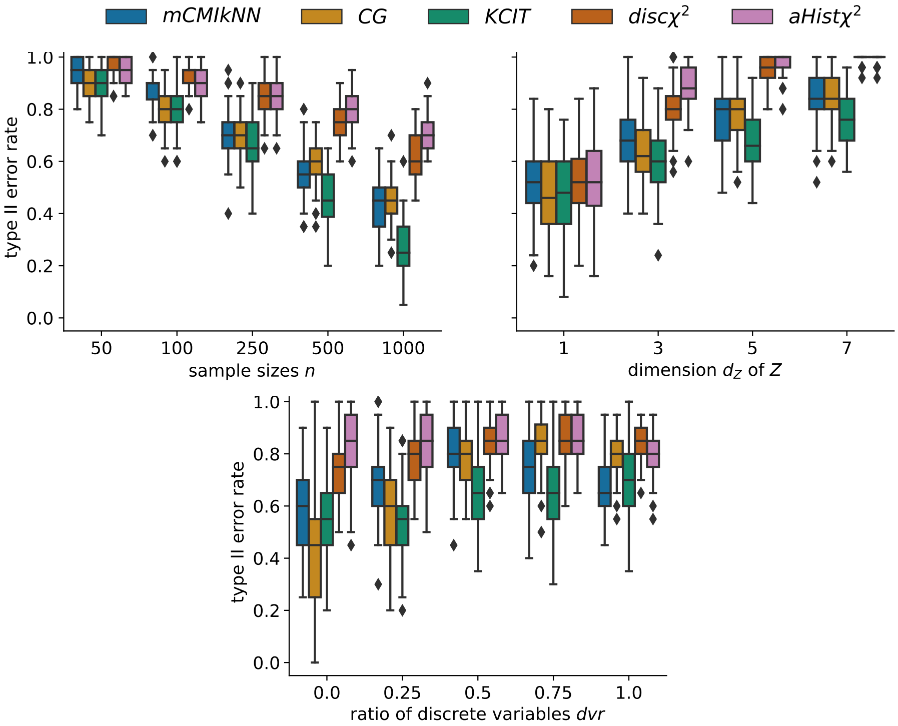


Fig. 9.4: Type II error rate (smaller better) of 20 000 CI decisions of the CI tests mCMIkNN, CG, KCIT, discχ^2, and aHistχ^2 with varying sample sizes n (top left), sizes of discrete domains d_Z (top right), and ratios of discrete variables dvr (bottom)[9].

stricted number of samples $n \leq 1\,000$, discχ^2 and aHistχ^2 suffer from the curse of dimensionality, which yields high type II error rates. For the continuous case, the linearity assumption of CG approximately covers some dependencies such that it achieves lower type II error rates. In contrast, for the discrete case, CG suffers from the combination of high degrees of freedom in combination with low sample sizes (similar to discχ^2 and aHistχ^2), in particular for high d_Z which yields high type II error rates.

In summary, mCMIkNN achieves statistical validity, i.e., robustness regarding type I errors, and power, i.e., robustness regarding type II errors, even for low sample sizes, which supplements the theoretically derived asymptotic results of Theorem 1 (page 93) and Theorem 2 (page 97), respectively.

9.4 Runtime Comparison

Lastly, we compare the mean runtimes of the different methods for CI testing over 2 400 CI decisions. In this context, we restrict the runtime measurements to the execution of the CI tests, i.e., omitting any data preparation, such as discretization in the case of discχ^2. Furthermore, we performed each CI test single-threaded on a server system with an Intel® Xeon® E7-4850 v4 CPU. Moreover, due to the long runtime of aHistχ^2, we limit the execution time to 600 seconds. In particular, we compare the runtimes for varying sample sizes n, dimensions of the conditioning set d_Z, and discrete variables ratios dvr in Table 9.4, Table 9.5, and Table 9.6, respectively.

Examining the CI tests' runtimes for increasing sample sizes n (see Table 9.4) shows expected behavior according to the methods' general computational complexity. For example, CG and discχ^2 achieve the fastest runtimes with fractions of seconds, even for a high number of samples[10]. The adaptive histogram-based aHistχ^2 suffers the longest runtimes for all sample sizes, which is also due to the curse of dimensionality yielding a worse performance for high-dimensional conditioning sets (see Table 9.5). While KCIT achieves a fast execution for small sample sizes, its cubic complexity yields long runtimes for high sample sizes. In contrast, mCMIkNN's log-linear complexity achieves a reasonable performance for small sample sizes and outperforms KCIT for sample sizes $n \geq 2\,000$.

Table 9.4: Mean runtimes in seconds (smaller better) of 2 400 CI decisions of mCMIkNN ($M_{perm} = 100$), CG, KCIT, discχ^2, and aHistχ^2 for an increasing number of samples $n \in \{50, 100, 250, 500, 1\,000, 2\,000\}$ (each of which covers different sizes of conditioning sets $d_Z \in \{1, 3, 5, 7\}$ and discrete variables ratios $dvr \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$).

Method \ n	50	100	250	500	1 000	2 000
mCMIkNN	0.411	0.914	2.929	7.713	20.995	58.109
CG	0.029	0.044	0.077	0.115	0.173	0.274
KCIT	0.014	0.035	0.255	1.648	12.114	102.012
$disc\chi^2$	0.001	0.001	0.001	0.001	0.001	0.001
$aHist\chi^2$	119.327	165.654	179.724	180.526	178.684	201.159

[10] Note that discχ^2 requires the discretization of continuous variables, which is excluded in all runtime measurements.

Table 9.5: Mean runtimes in seconds (smaller better) of 2 400 CI decisions of `mCMIkNN` ($M_{perm} = 100$), `CG`, `KCIT`, `disc`χ^2, and `aHist`χ^2 for increasing sizes conditioning sets $d_Z \in \{1, 3, 5, 7\}$ (each of which covering different sample sizes $n \in \{50, 100, 250, 500, 1\,000, 2\,000\}$ and discrete variables ratios $dvr \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$).

Method \ d_Z	1	3	5	7
mCMIkNN	7.849	9.887	17.474	25.505
CG	0.006	0.032	0.144	0.292
KCIT	19.425	19.249	19.313	19.397
discχ^2	0.001	0.001	0.001	0.002
aHistχ^2	2.214	62.267	307.724	311.178

Regarding an increase of the dimensionality of the conditioning set d_Z (see Table 9.5), the runtimes indicate that `aHist`χ^2 struggles strongly in high-dimensional settings. Similarly, but to a lesser extent, the runtimes of `CG` and `mCMIkNN` increase with the conditioning sets' size. In contrast, `disc`χ^2 and `KCIT` achieve stable runtimes for all dimensions[10].

For varying discrete variables ratios dvr (see Table 9.6), `disc`χ^2 and `KCIT` achieve stable runtimes for all settings, too[10]. In contrast, the runtimes of `mCMIkNN` and `CG` increase for increasing discrete variables ratios, and `aHist`χ^2 struggles in mixed cases with $dvr \in \{0.25, 0.5, 0.75\}$. In the case of `mCMIkNN`, the poorer performance for higher discrete variable ratios dvr is caused by the use of *k*-*d* trees when computing the kNN, as *k*-*d* trees are less efficient for discrete data, compared to continuous data.

Note that the permutation scheme of `mCMIkNN` is well suited for parallel execution strategies to speed up the computation. In particular, the computational expensive M_{perm} permutations in Algorithm 2 (on page 93) can be embarrassingly parallelized, e.g., see [148]. Further, recent research on hardware acceleration has shown that kNN-based CI tests can be efficiently executed on GPUs, particularly when used in constraint-based causal discovery [55]. For a comprehensive examination of GPU-accelerated constraint-based causal discovery, we refer to the doctoral thesis of Christopher Hagedorn [51].

Table 9.6: Mean runtimes in seconds (smaller better) of 2 400 CI decisions of `mCMIkNN` ($M_{perm} = 100$), `CG`, `KCIT`, `disc`χ^2, and `aHist`χ^2 for increasing discrete variables ratios $dvr \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$ (each of which covering different sample sizes $n \in \{50, 100, 250, 500, 1\,000, 2000\}$ and sizes of conditioning sets $d_Z \in \{1, 3, 5, 7\}$).

Method \ dvr	0.0	0.25	0.5	0.75	1.0
mCMIkNN	12.554	11.677	13.067	17.938	20.656
CG	0.002	0.02	0.091	0.291	0.189
KCIT	19.615	19.33	19.257	19.379	19.151
discχ^2	0.001	0.001	0.001	0.001	0.002
aHistχ^2	21.506	138.145	181.078	328.064	185.435

9.5 Causal Discovery

In this section, we evaluate how the accuracy of the considered CI tests affects the consistency of constraint-based causal discovery. Therefore, we use the CI tests in the PC-stable algorithm [20] to estimate the CPDAG $\mathcal{G}_{CPDAG}$ of the DAG $\mathcal{G}$ on data generated according to Section 9.1 on page 103. In particular, we evaluate F1 (Section 9.5.1) and ROC AUC scores (Section 9.5.2).

Note that we excluded `aHist`χ^2 in the evaluation of causal discovery as its long execution time and restricted implementation to non-discrete data do not allow for usage in constraint-based causal discovery.

9.5.1 F1 Scores of Causal Structures

In Fig. 9.5, we examine the F1 scores [16] of erroneously detected edges in the skeletons of $\hat{\mathcal{G}}_{CPDAG,n}(0.05)$ estimated with PC-stable using the respective CI tests in comparison to the true skeleton of $\mathcal{G}$. In this context, we choose $\alpha = 0.05$, which takes the more complex CI characteristics present in causal discovery (e.g., confounders, colliders, paths) into account, such that $M_{perm} = 100$ is a sufficient choice for `mCMIkNN`.

While F1 grows for all methods as n increases, `mCMIkNN` outperforms the competitors (Fig. 9.5 top left). Further, `mCMIkNN` achieves the highest F1 scores for high discrete variables ratios (Fig. 9.5 top right). Note that `disc`χ^2 achieves high F1 scores due to the sparsity of samples CGMs, which yields low conditioning sizes, where `disc`χ^2 has small type I error rates (see Fig. 9.3 top right) and is still able to control type II errors (see

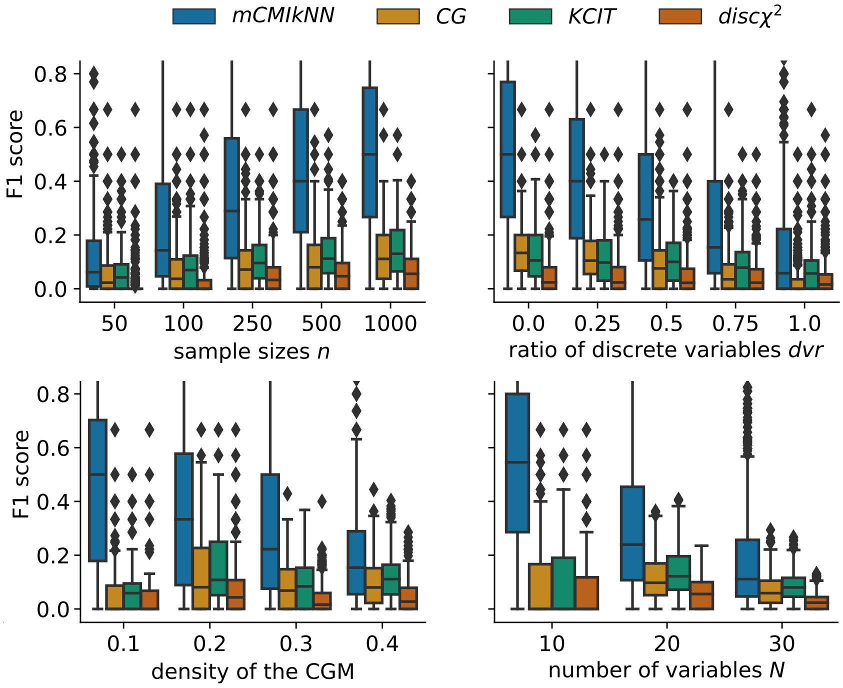


Fig. 9.5: F1 scores (higher better) of PC-stable with CI tests `mCMIkNN`, `CG`, `KCIT`, and `disc`χ^2 computed over 3 000 CGMs for varying the sample sizes n (top left), discrete variable ratios (top right), densities of CGMs (bottom left), and numbers of variables (bottom right).

Fig. 9.4 top right). In this context, note that F1 scores are balanced towards type I error rates which are crucial in causal discovery. Particularly as the considered directed acyclic graph (DAG) are sparse, i.e., have a relatively small number of edges compared to the set of possible edges (which is N^{N-2}). In this case, the number of possible type I errors exceeds the number of possible type II errors. Hence, the rates should be considered imbalanced with a focus on type I error rates. Further, constraint-based causal discovery starts with a fully connected graph and iteratively deletes edges with increasing size of conditioning sets for each level d_Z according to adjacent nodes. Hence, type I errors are carried on to higher levels d_Z, while type II errors are more balanced as for increasing d_Z more edges are deleted due to the curse of dimensionality. Further, constraint-based causal discovery requires higher sample sizes for consistency due to the multiple testing problem [44, 168].

All methods suffer from the curse of dimensionality, i.e., a decreasing F1 score for increasing densities (Fig. 9.5 bottom left) and numbers of variables (Fig. 9.5 bottom right) which yields larger conditioning sizes d_Z.

9.5.2 ROC AUC Scores of Causal Structures

To provide a complete examination, we also present the ROC AUC scores of wrongly detected edges in the skeletons of $\hat{\mathcal{G}}_{CPDAG,n}(0.05)$ estimated with PC-stable using the respective CI tests in comparison to the true skeleton of $\mathcal{G}$. Examining type I and type II errors more balanced using ROC AUC scores (Fig. 9.6) shows no noteworthy differences.

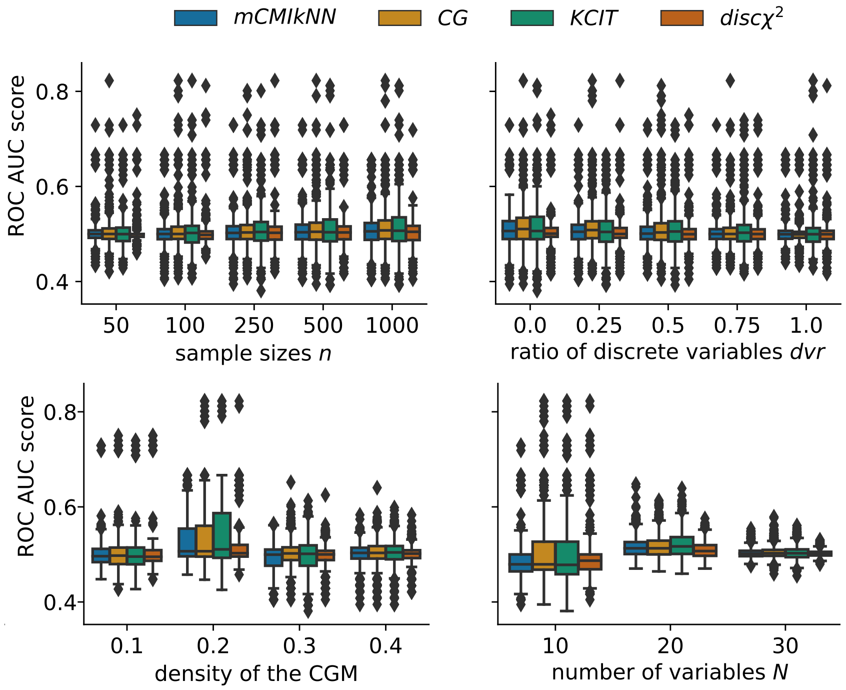


Fig. 9.6: ROC AUC scores (higher better) of PC-stable using CI tests `mCMIkNN`, `CG`, `KCIT`, and `disc`χ^2 computed over 3 000 CGMs for varying the sample sizes n (top left), discrete variable ratios (top right), densities of CGMs (bottom left), and numbers of variables (bottom right).

10

Application Scenario in Discrete Manufacturing

In this chapter, we introduce the real-world industrial manufacturing scenario and demonstrate that `mCMIkNN` outperforms commonly used discretization-based approaches for constraint-based causal discovery in practice. In particular, we motivate the use-case and sketch further challenges (Section 10.1). Further, we describe the simplified manufacturing scenario and an empirical evaluation of `mCMIkNN`-based causal discovery (Section 10.2). We complete our real-world scenario with concluding remarks, point out limitations, and discuss future work (Section 10.3).

Contribution: Parts of this chapter have previously been published in the journal paper [69]. The thesis author established the cooperation with the industry partner and conducted the experimental evaluation supported by Christopher Hagedorn. The applied methods are based on the thesis author's theoretical concepts as described in the previous chapters, and the thesis author prepared the original draft. The coauthors improved the paper's material and its presentation.

10.1 Product Quality in Discrete Manufacturing

In this section, we motivate our real-world scenario that has been examined together with our cooperation partner from discrete manufacturing (Section 10.1.1), provide background information on the production process (Section 10.1.2), and discuss goals as well as challenges (Section 10.1.3).

10.1.1 Motivation

Modern discrete manufacturing enterprises face growing demands for increased product quality, diversified products that collide with shortened product life-cycles, reduced costs, and global competition [97]. In this context, production quality performance during the machinery production process is of fundamental relevance [15, 176]. Therefore, enhancing productivity and effective quality improvement can be instrumental in increasing an enterprise's competitive power [119]. Moreover, the machinery's configurations have a profound impact on the performance of the manufacturing system in terms of productivity, product quality, capacity, scalability, and costs [81]. Therefore, understanding causal structures between machinery configurations, product characteristics, and the respective product quality are essential for enhancing the productivity of the manufacturing process [1]. On the other hand, understanding causal structures in the industrial domain usually relies on domain knowledge and intuition as industrial manufacturing processes become more complex, with sometimes hundreds of possible factors [108]. In this context, the emergence of methods for causal discovery creates the basis for attempts of a data-driven assessment of the causal structures from observational data of manufacturing processes, e.g., see [54, 70, 108].

For more information and related work on causal discovery for root cause analysis (RCA) in discrete manufacturing, we refer to the comprehensive examination in Chapter 5 on page 57.

10.1.2 Background on the Machinery Production Process

A typical machinery production process can be schematically divided into the *configuration phase* and the *production phase* as sketched in Fig. 10.1.

While the production process is highly automated, the configuration phase requires substantial human participation as the machinery technician has to configure the machinery settings. In particular, the machinery has to be properly configured to ensure a minimum of rejected goods that do not meet required quality targets. Therefore, the technician aims to start the machinery production process in the configuration phase to obtain the quality targets through an iterative adaption of possible machinery configuration settings such as speed. In this context, all products that do not meet the quality targets are rejected. In case quality standards are met, machinery configurations S_{con}, achieved quality results Q_{con}, and the number of rejected goods R_{con} within the configuration phase are logged. Then, the discrete production process enters the production phase with a high throughput of produced products over several units of similar design using the derived machinery configurations. Finally, the number of rejected goods within the production phase R_{prod} is logged, too.

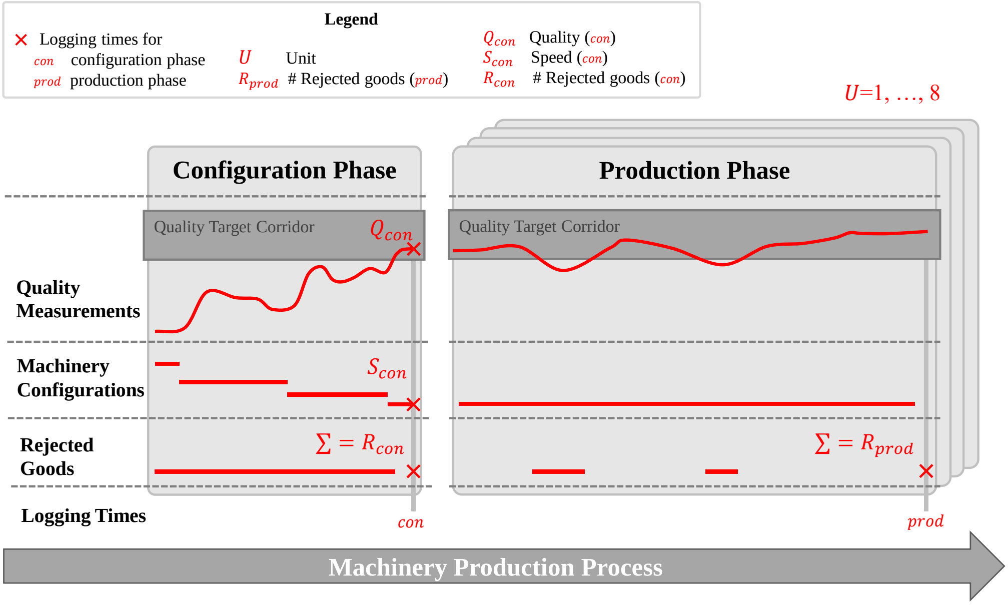


Fig. 10.1: A schematic overview of a machinery production process, where machinery configurations, e.g., speed S_{con}, are adapted within the *configuration phase* to obtain the required quality target for a quality measurement Q_{con} avoiding rejected goods R_{con}. This configuration aims to reduce the number of rejected goods R_{prod} within a high throughput *production phase* over several units U.

10.1.3 Goal and Challenges

As causal structural knowledge serves as the basis for data-driven decision support, e.g., see [54, 70, 108], we aim to estimate the underlying causal structures of the above-described discrete manufacturing process. In practice, we face the following well-known challenges that have previously been introduced in Section 5.1.2 on page 58, too.

- *Challenge I (High-Dimensionality):* A machine logs its configuration parameters, internal state based on sensor readings, and error messages during production. Thus, the machinery raw data contains millions of entries with several thousand different types, resulting in high-dimensional data. High-dimensional data leads to long execution times, hindering the application of causal discovery in practice [89]. Further, it increases the potential for statistical error [104].;
- *Challenge II (Semi-Structured Raw Data):* The data is recorded at different time intervals and may be stored in a semi-structured log format. Hence, the semi-structured machinery raw data needs to be preprocessed [188] before the application of causal discovery to extract the independent and identically distributed (i.i.d) observational samples [172].
- *Challenge III (Mixed Discrete-Continuous Data):* The machinery raw log data contains a mixture of continuous variables, such as sensor measurements, and discrete variables, such as configuration parameters or error messages such that mixed discrete-continuous data is common in practice, e.g., see [54, 70].

To allow for a comprehensive real-world example, see *Challenge I*, we restrict our attention to the main factors within the production process proposed by domain experts. In particular, we consider the variables described in the schematic overview in Fig. 10.1.

In this context, the manufacturing of one good in the production phase can be described by the variables Q_{con}, S_{con}, and R_{con} determined in the configuration phase, and R_{prod} as well as the locality, i.e., unit U, the good is produced. Further, we apply the transformation rules proposed introduced in Section 5.4.1 on page 65 and leverage the knowledge of domain experts to receive sound observational data, see *Challenge II* [54].

On this basis, we evaluate the accuracy of `mCMIkNN` in mixed discrete-continuous real-world data compared to commonly applied discretization-based approaches for causal discovery, see *Challenge III.*

10.2 Evaluation of the Discrete Manufacturing Scenario

In this section, we compare the accuracy of `mCMIkNN` against the commonly used discretization-based approach of causal discovery to derive the underlying causal structures of the above-described discrete manufacturing scenario. Therefore, we shortly provide an overview of the data (Section 10.2.1), the assumed underlying DAG (Section 10.2.2), and estimate the CPDAGs from the real-world data (Section 10.2.3).

10.2.1 Observational Data of the Manufacturing Process

In line with the manufacturing process described in Section 10.1, we consider the continuous variables Q_{con}, S_{con}, R_{con}, and R_{prod} (which may follow a mixture distribution), as well as the discrete U. The transformation of the semi-structured log data that has been proposed in Section 5.4.1 (page 65) yields approximately 1 300 sound samples of the discrete manufacturing process, each associated with one produced good with thousands of pieces made in the production phase. In line with all experiments in this part of the thesis, we max-min normalize continuous variables.

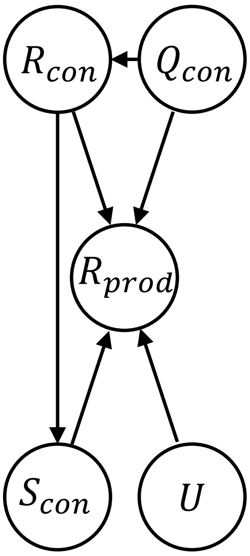


Fig. 10.2: True underlying DAG.

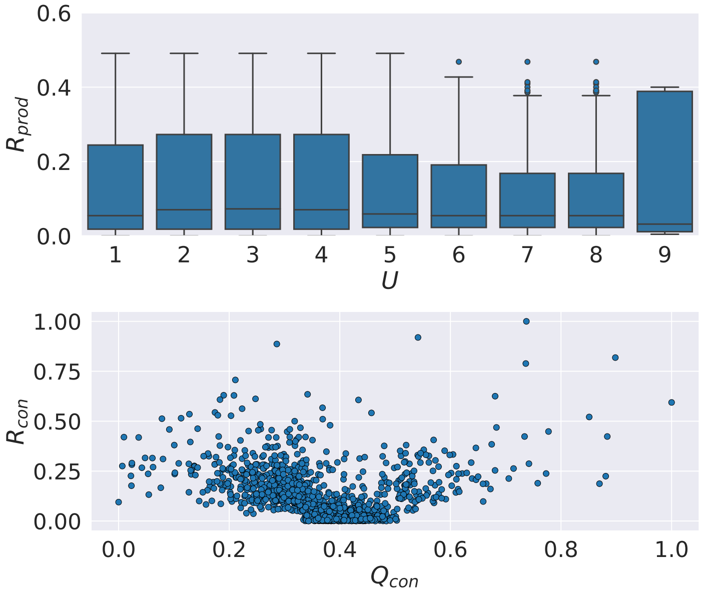


Fig. 10.3: Real-world data characteristics for $U \rightarrow R_{prod}$ (top) and for $Q_{con} \rightarrow R_{prod}$ (bottom).

10.2.2 Assumed Ground-Truth Causal Structures

With the help of domain experts from our cooperation partner, we define the DAG $\mathcal{G}$ depicted in Fig. 10.2 that serves as ground truth representing the underlying causal structures of the discrete manufacturing process.

The causal structures arise as follows. Quality measurements Q_{con} and rejections R_{con} are used for adjustment of the processing speed S_{con} in the configuration phase with respective edges $Q_{con} \rightarrow R_{con}$, $R_{con} \rightarrow S_{con}$. Then, for all units U, the high throughput production process is started according to the configuration. Hence, we assume the following edges hold true $Q_{con} \rightarrow R_{prod}$, $R_{con} \rightarrow R_{prod}$, $S_{con} \rightarrow R_{prod}$, and $U \rightarrow R_{prod}$.

Our small real-world scenario covers the omnipresent characteristics of mixed discrete-continuous data in discrete manufacturing, see [54]. For example, as depicted in Fig. 10.3, our data contains the mixed discrete-continuous relationship $U \rightarrow R_{prod}$ (Fig. 10.3 top) and the non-linear relationship $Q_{con} \rightarrow R_{prod}$ (Fig. 10.3 bottom).

In this context, note that we cannot guarantee causal faithfulness, i.e., there may exist confounders not considered within our small example.

10.2.3 Evaluation

In line with the experiments on constraint-based causal discovery in Section 9.5 (page 114), we apply the PC-stable algorithm ($\alpha = 0.05$) to estimate the complete partially directed acyclic graph (CPDAG) $\mathcal{G}_{CPDAG}$ of the true directed acyclic graph (DAG) $\mathcal{G}$. In this context, we compare `mCMIkNN` ($k_{perm} = 5$, $k_{CMI} = 25$, $M_{perm} = 100$) against the commonly applied discretization-based CI test `disc`χ^2. For more information on `disc`χ^2 or on the parameters of `mCMIkNN`, we refer to Section 9.3 or Section 8.4.5, respectively.

As depicted in Fig. 10.4, the CPDAG estimated with PC-stable using `mCMIkNN` (Fig. 10.4 right) is closer to the assumed true DAG (Fig. 10.4 left) compared to the

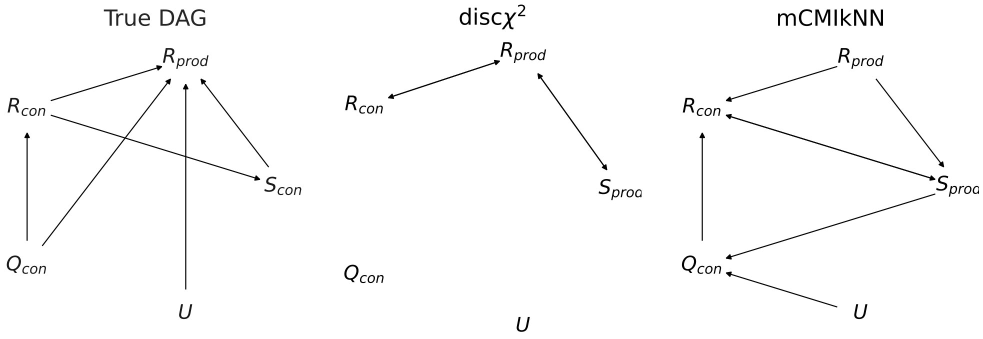


Fig. 10.4: Assumed true DAG (left) and the estimated CPDAGs using the PC-stable algorithm with `disc`χ^2, F1 $= 0.40$ (center), and `mCMIkNN`, F1 $= 0.57$ (right).

CPDAG estimated with PC-stable using `disc`χ^2 (Fig. 10.4 center). The performance difference is reflected in the F1 scores calculated on the respective skeletons, i.e., F1 $= 0.57$ for `mCMIkNN` vs. F1 $= 0.4$ for `disc`χ^2.

Therefore, the empirical results indicating the capabilities of `mCMIkNN` to capture complex causal mechanisms in mixed discrete-continuous data (see Section 9.3 on page 108) transfer to real-world data, too.

10.3 Discussion

In this section, we summarize the results of the real-world scenario (Section 10.3.1), and point out limitations (Section 10.3.2).

10.3.1 Summary

We demonstrated that `mCMIkNN` captures the complex underlying causal mechanisms and outperforms the commonly used discretization-based approach for constraint-based causal discovery in a real-world discrete manufacturing scenario.

10.3.2 Limitations

In this context, note that the accuracy of the estimated CPDAGs is affected by latent confounding variables not present in the data. In particular, within many cycles with domain experts, we extended the small scenario to cover other driving factors within the machinery production process. In this context, a similar behavior was visible within more complex graphs of up to 50 variables where `mCMIkNN` outperformed `disc`χ^2, too. Overall, `mCMIkNN` captured the CI characteristics of the mixed discrete-continuous discrete manufacturing data compared to `disc`χ^2. For a detailed discussion on limitations of causal discovery in practice, we refer to Section 5.5.2 on page 75. Moreover, for more information on challenges of causal discovery in practice and constraint-based methods that allow for latent variables, we refer to [44, 54, 106], and [145, 168, 174], respectively.

11

Conclusion of Part II

In this chapter, we close Part II by summarizing our results (Section 11.1), point out limitations (Section 11.2), and discuss future work (Section 11.3).

Contribution: Parts of this chapter have previously been published in the paper [69]. The thesis author's detailed contributions are discussed at the beginning of the chapters Chapter 8, Chapter 9, and Chapter 10, respectively.

11.1 Summary

Recap that this second part of this thesis, Part II, was motivated with the goal of weakening the assumptions of constraint-based causal discovery on the data characteristics (see Section 2.1 on page 25). In this context, we contributed in a fourfold manner.

- We proposed a kNN-based local CP scheme to derive a non-parametric CI test, called `mCMIkNN`, using a kNN-based CMI estimator as a test statistic.
- We provided theoretical results on the CI test's validity and power. In particular, we proved that `mCMIkNN` can control type I and type II errors.
- We showed that `mCMIkNN` allows for consistent estimation of causal structures when used in constraint-based causal discovery.
- An extensive evaluation on synthetic and real-world data showed that `mCMIkNN` outperforms state-of-the-art competitors, particularly for low sample sizes.
- We showed that `mCMIkNN` improves the accuracy of causal discovery in a real-world discrete manufacturing scenario.

In the following, we explain the added value of the above contributions regarding an improved applicability of methods for causal discovery in practice.

Non-Parametric CI Testing: The development of `mCMIkNN` as a non-parametric CI test allows to state mild assumptions on the data characteristics, see **(R1)**.

In particular, in case non-singularity holds, *(A1)* and *(A2)* is satisfied whenever the data distribution is (i) (finitely) discrete, (ii) continuous, (iii) some dimensions are (countably) discrete and some are continuous, and (iv) a mixture of the previous cases, which covers most real-world data [43, 112]. Hence, application in practice reduces to checking the sufficient condition of non-singularity, which can be done by ensuring that there exists no set $E \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ such that $P_{XY|Z}(E) = 0$ while $P_{Y|Z}(E_a) = 0$ for $E_b = \{(x, z) : (x, b, z) \in E\}$ and $E_a = \{(y, z) : (a, y, z) \in E\}$, see [42, 199].

Therefore, `mCMIkNN` enables causal discovery under mild assumptions, which can be ensured according to the methods proposed in [42, 199].

Validtiy and Power: Our asymptotic results on mCMIkNN's statistical validity (see Theorem 1 on page 93) and power (see Theorem 2 on page 97) ensure its theoretical correctness. Furthermore, the asymptotic consistency of mCMIkNN-based causal discovery (see Theorem 3 on page 98) guarantees the accuracy of the learned CPDAGs for large sample sizes.

Synthetic and Real-World Evaluation: Our experimental evaluation on synthetic data supplements our theoretical results, indicating that mCMIkNN is robust regarding type I and type II errors in the finite case, too. In particular, we showed that mCMIkNN outperforms state-of-the-art competitors, particularly for low sample sizes. Moreover, we showed that these results transfer to real-world data, which further demonstrates the applicability of mCMIkNN in practice.

11.2 Limitations

While the mild assumptions *(A1)* and *(A2)* simplify the application of mCMIkNN in practice, we cannot derive bounds on type I and II error control for the finite case as provided in [78], but the empirical results showed that mCMIkNN is robust in the finite case, too. Note that these bounds can be achieved by considering stronger assumptions, such as lower bounds on corresponding probabilities for discrete points, e.g., see [3, 78], or smoothness assumptions for continuous variables, e.g., see [7, 198].

Further, the current implementation of mCMIkNN is restricted to metric spaces and, hence, does not allow for categorical variables. To extend the implementation to categorical variables, an isometric mapping into the metric space can be examined, see [117].

Note that kNN methods are not invariant regarding the scaling of variables, and their computational complexity yields long runtimes, particularly in large sample sizes. In this context, our work on hardware acceleration has shown that the runtime of mCMIkNN can significantly be speeded up using GPUs, see **(R2)**.

As this thesis focuses on assumptions of CI tests that impede the application of constraint-based causal discovery in practice, see **RQ2**, we did not encounter violations of other assumptions necessary for constraint-based causal discovery. For example, we assume that faithfulness or causal sufficiency holds true. In this context, there is ongoing research on extensions of constraint-based methods that tackle these challenges to further improve the accuracy of causal discovery in practice, e.g., see [145, 168, 174] For an overview of existing methods, we refer to [168, 44, 193].

11.3 Future Work

In future work, we will examine the performance regarding more complex data-generating mechanisms and mixed data that incorporates categorical variables, too.

We consider parallel execution strategies to speed up the computation, e.g., parallelizing the execution of M_{perm} permutations in Algorithm 2, e.g., see [148], or using GPUs [55]. In this context, note that our research on hardware acceleration has shown that GPUs allow to speed up constraint-based causal discovery that uses a similar CI test [55] In particular, experiments showed that the runtime is mainly affected by the number of k-nearest-neighbors within the CMI estimation. Depending on the chosen number of k, we achieved a speedup up to 352 for a single CI test, which results in speedup factors for causal discovery of up to 1 000. For more information, we refer to [55] or see the comprehensive work of Hagedorn [51].

Part III

Causal Discovery in Practice and Conclusion

In the first two parts of this thesis, we provided tooling for causal discovery, see Part I, and developed a non-parametric CI test, see Part II. In particular, our tooling supports the evaluation of methods for causal discovery and their applicability in practice tackling **RQ1**. Further, the mild assumptions of `mCMIkNN` on data characteristics improve the accuracy of constraint-based causal discovery in practice tackling **RQ2**. Although the presented tools and methods provide a comprehensive toolkit, they require a deep understanding of the methods and concepts of causal discovery when used in practice. Therefore, it remains open to demonstrate that causal graphs can be efficiently integrated into the workflow of a machine operator.

In this closing part, we demonstrate how causal structural knowledge can be integrated into monitoring tools to provide decision support to machine operators. In particular, we propose to integrate the learned causal structures into existing monitoring solutions using the example of an automotive body shop assembly line.

Hence, together with Part I and Part II, we achieved a comprehensive examination of our two research questions such that we close this thesis by recapitulating our contributions regarding **RQ1** and **RQ2**.

12

An Automotive Manufacturing Use Case

In this chapter, we demonstrate how causal structural knowledge adds decision support in a real-world use case from automotive manufacturing. In this context, online monitoring, and failure diagnosis, also under the prism of improving root cause analysis (RCA), is of great importance (Section 12.1). We showcase how causal graphs can be incorporated into a monitoring tool to function as a decision support system for an operator of a modern automotive body shop assembly line and enable fast and effective handling of failures and quality deviations (Section 12.2).

Contribution: Parts of this chapter have previously been published in the demonstration paper [70]. The monitoring tool was developed with seven undergraduate students as part of their final project. Hagedorn and the thesis author, established the cooperation with the industry partner, developed the monitoring tool's concept, and guided the implementation, which was almost entirely handled by the students. Furthermore, the thesis author worked out the application concepts of causal discovery, ensured the correctness of the applied mathematical concepts, and prepared the original draft. The coauthors improved the paper's material and its presentation.

12.1 Motivation for Causally-Enriched Monitoring Systems

Automotive manufacturing enterprises have to cope with growing demands for increased product quality, greater product variability, shorter product life cycles, reduced cost, and global competition [97]. In order to meet these demands, modern car body shop assembly lines are highly optimized and operative with a minimum of human intervention. Hence, the occurrence of failures and deviations of quality measurements that require human intervention are a major cause of unscheduled stoppage of the car body assembly line and are costly not only in terms of time lost but also in terms of capital destroyed [19]. Therefore, as depicted in Fig. 12.1, monitoring systems for the operator of an industrial plant have the intention to cover the current status of the assembly line including the involved car bodies, and occurring failures to ensure a fast reaction in case of interruptions [197].

In case of an interruption of the production process, the RCA of failures and quality deviations is usually built upon non-persistent, individual on-site expert knowledge, and hence troubleshooting relies on the individual knowledge of the staff on shift. For an exemplary use case that covers the RCA of quality deviations, we refer to our welding process example that motivated this thesis (Section 1.1 on page 1). Advances of data-driven machine learning techniques have opened the possibilities to create monitoring applications that integrate failure diagnosis [1, 102]. Moreover, the emergence of methods for causal discovery, e.g., see [168, 130], created the basis for attempts of a purely

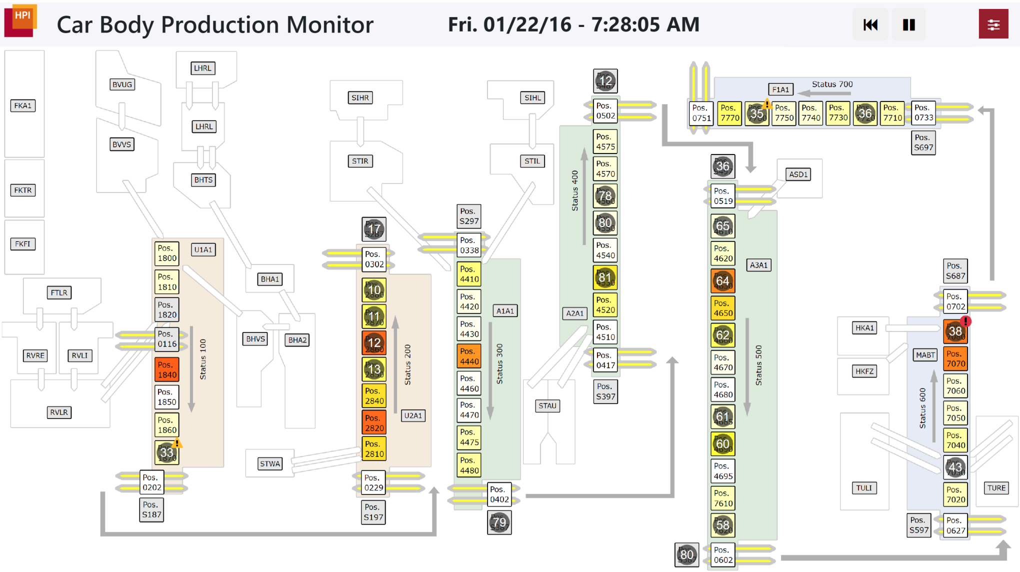


Fig. 12.1: *Monitoring View* depicting the vehicles (dots) within the different production cells of the car body shop assembly line.

data-driven assessment of the causal structures from observational data of a manufacturing process [95, 108]. For a gentle introduction to causal discovery, also regarding the simplified welding process, see Section 1.2 on page 3.

While previous work addresses challenges in domain-specific data preprocessing and examines the quality of the derived causal structures, e.g., see Chapter 5 on page 57, it remains open to demonstrate how the knowledge about causal structures can be leveraged within the assembly line. Consequently, a demonstration of the possibilities of incorporation of causal graphs into the assembly lines for monitoring, and failure diagnosis, also under the prism of making troubleshooting more efficient, is of great importance.

12.2 How Causal Structural Knowledge Adds Decision Support

In this section, we provide some background on the causal modeling of an assembly line (Section 12.2.1) before demonstrating the added-value of a causally enriched monitoring tool (Section 12.2.2).

12.2.1 Causal Modeling of the Assembly Line

In the following, we sketch the steps for learning the causal structures from historic raw log data and for failure prediction. For a detailed examination of all steps and the applied concepts, we refer to Section 5.4 on page 64.

Production Process: An automotive body shop assembly line consists of multiple sections, e.g., responsible for attachments where the prefabricated individual parts are fitted to the car's body. Each section is separated into high-automated production cells in which robots weld, rivet, or bend summing up to several hundreds of involved robots constantly streaming data, e.g., failures or quality measurements.

Data Preprocessing: The raw data comprises historic logged failure occurrences and quality measures, categorized by their severity, of a two-year production time for one car body type. An iterative approach incorporating process metadata under consideration of domain knowledge yields 70k sound discrete observations of 6.5k variables, e.g., according to [54]. Each observation covers the failure and quality deviation history of a specific car body.

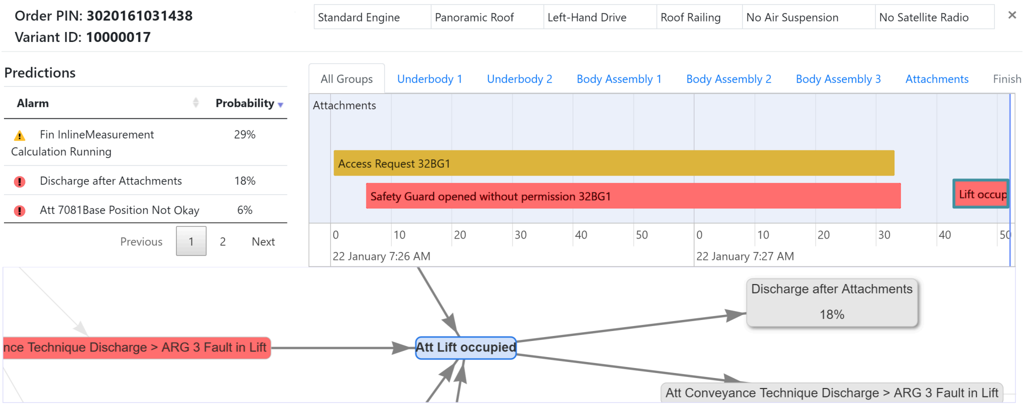
Causal Discovery: Given the historic observational data, the causal structures are learned through constraint-based causal discovery using the PC algorithm taking path constraints implied by domain knowledge into account, e.g., see [168, 9, 54]. Hence, the edges between nodes represent causal relationships between failures or quality deviations whereof focused selections were evaluated by domain experts.

Failure Prediction: The parameterized causal graph, e.g., see [121], enables calculation of the conditional probability of failures and critical quality deviations given the current state of the car body's production process.

12.2.2 A Causally-Enriched Monitoring Tool

In this section, we showcase the application possibilities of our on-line monitoring tool that incorporates the causal knowledge between failures and quality deviations and demonstrate the opportunities concerning more efficient troubleshooting.

As the occurrence of failures and quality deviations interrupting the highly automatized production process requires an instant human intervention of the technical staff it is essential to have an accurate view of the current state of the body shop assembly line. Hence, our *Monitoring View*, depicted in Fig. 12.1, provides an entry point to the main sections of the assembly line where the currently produced car bodies are depicted within the production cells. While the darker highlighted production cells indicate an increased probability of critical interruptions, and hence should be called under the attention, the occurrence of warning signs at the car bodies directly refers to the occurrence of failures or quality deviations requesting an instant manual troubleshooting.



Order PIN: **3020161031438**
Variant ID: **10000017**

Standard Engine | Panoramic Roof | Left-Hand Drive | Roof Railing | No Air Suspension | No Satellite Radio

Predictions

Alarm	Probability
⚠ Fin InlineMeasurement Calculation Running	29%
❗ Discharge after Attachments	18%
❗ Att 7081Base Position Not Okay	6%

Fig. 12.2: *Detailed Failure View* of a car body including failure history (top right), individual failure prediction (top left), and corresponding causal graphical model (bottom) given the current failure in the center with possible root causes to the left and subsequent failures to the right.

In this situation, where every minute of an unscheduled stopping results in loss of money the *Detailed Failure View*, see Fig. 12.2, provides the causal graph depicting all possible preceding root causes to the current failure - highlighted in blue - as well as the full failure history of the affected car body. The causal graph incorporates possible root causes for the whole assembly line and additionally refers to actually occurred, yet oftentimes unnoticed, root causes - vertices highlighted in red. Thus, the technical staff receives treatment recommendation that outperforms the usually non-persistent, individual on-site expert knowledge. Moreover, given the currently observed failure the depiction of possible subsequent critical failures in the causal graph, with their probability estimation, enables and guides the technical staff within a predictive troubleshooting. In order to provide a comprehensive examination of possible future failures and quality deviations, a list covers critical failure predictions based on the knowledge about the causal structures and the car body's full history of occurred failures. This combination of early warning in both the *Monitoring View* and the *Detailed Failure View*, the direct identification of root causes through the incorporation of a causal graph, and the extension through the prediction of subsequent failures builds the basis of a data-driven decision support for on-line monitoring of automotive body shop assembly lines.

12.3 Discussion

In this chapter, we demonstrated the possibilities of an integration of causal structural knowledge into a modern automotive body shop assembly line. Thereby, we enriched a monitoring tool with a causal graph representing the causal structures between failures and quality deviations of the production process to provide the operator of an assembly line with data-driven decision support enabling fast and effective troubleshooting.

This demonstration supplements the previous parts and closes the gap between our contributions to improving causal discovery and its real-world application to provide root cause analysis in the workflow of a machine operator. Hence, we close the loop to our exemplary discrete manufacturing process and the machine operator aiming to improve quality deviations of a welding process within an automotive body shop assembly line (see Section 1.1 on page 1).

13

Conclusion of the Thesis

In this closing chapter, we conclude our work and respond to initial research questions **RQ1** (Section 13.1) and **RQ2** (Section 13.2).

13.1 Tooling for Causal Discovery

As the knowledge of underlying causal structures is the basis for decision support, causal discovery has received widespread attention, e.g., see [130, 70, 54]. In recent years, the corresponding research addressing challenges of causal discovery in practice has led to a broad spectrum of different methods and implementations, each having specific assumptions and accuracy characteristics or is introducing implementation-specific overhead in the runtime. Hence, methods for causal discovery should be validated within different scenarios, including a varying number of variables or sensitivity of parameters, aiming to understand the method's behaviors in specific edge cases, e.g., when underlying assumptions are violated [37, 85, 44, 67]. However, considering and evaluating a selection of algorithms or different implementations in different programming languages utilizing different hardware setups becomes a tedious manual task with high setup costs [68, 5].

To tackle the aforementioned challenges, we formulated our first research question (see Section 1.3.1) that was answered in this thesis.

- **Research Question 1 (RQ1):** *How to support the evaluation of methods for causal discovery and their applicability in practice?*

To provide a platform-independent modular pipeline for causal discovery with comprehensive evaluation opportunities, we presented `MPCSL` (Chapter 3). In this context, we demonstrated the capabilities of `MPCSL` within a case study, where we evaluate existing implementations of the well-known PC algorithm concerning their runtime performance characteristics. Further, we introduced the mixed additive noise model (MANM) that provides a ground truth model for generating observational data following various distribution models and present our ground truth framework `MANM-CS` (Chapter 4). In particular, we demonstrated the usability of `MANM-CS` compared to well-known benchmark data sets and in a simple benchmarking experiment on the accuracy of causal discovery from mixed discrete-continuous data. Finally, we demonstrated that these tools support causal discovery from manufacturing log data to understand unforeseen production downtimes (Chapter 5). In particular, we showcased challenges and requirements that arise when dealing with raw log data and provided necessary concepts for the transferability of causal discovery to practice.

13.2 Causal Discovery from Mixed Discrete-Continuous Data

Testing for conditional independence (CI) is a fundamental task for constraint-based causal discovery but is particularly challenging in mixed discrete-continuous data omnipresent in many real-world scenarios [44, 168, 106]. In this context, inadequate assumptions or discretization of continuous variables reduce the CI test's statistical power, which yields incorrect learned causal structures [29, 139].

To tackle the aforementioned challenges, we formulated our second research question (see Section 1.3.2) that was answered in this thesis.

- **Research Question 2 (RQ2):** *How to weaken the assumptions of constraint-based causal discovery on data characteristics?*

To derive weaker assumptions, we proposed a kNN-based local conditional permutation scheme to derive a non-parametric CI, called `mCMIkNN` (Section 8.4.2). In particular, `mCMIkNN` enables constraint-based causal discovery under mild assumptions on the data characteristics. We provided theoretical results on the CI test's validity and power (Section 8.4.3). In particular, we proved that `mCMIkNN` is able to control type I and type II errors. Further, we proved that `mCMIkNN` allows for consistent estimation of causal structures when used in constraint-based causal discovery. An extensive evaluation on synthetic data supplemented our theoretic results and showed that `mCMIkNN` outperforms state-of-the-art competitors, particularly for low sample sizes (Chapter 9). Finally, we demonstrated `mCMIkNN`' capabilities to capture complex causal mechanisms in mixed discrete-continuous data transfer to real-world data too, which further demonstrates the applicability of `mCMIkNN` in practice (Chapter 10).

In view of our work on two challenges of causal discovery in practice, `MPCSL` and `MANM-CS` provide researchers and practitioners tools to evaluate and apply methods for causal discovery considering various application scenarios. Additionally, the non-parametric CI test `mCMIkNN` states mild assumptions on data characteristics, hence improving the accuracy of constraint-based causal discovery in practice. Furthermore, our real-world use cases demonstrate that our achievements can be transferred to practice yielding data-driven decision support in various application scenarios. Therefore, we take two crucial steps to improve the applicability of causal discovery in practice.

A

Appendix

The following appendix complements the information provided in this thesis. In Appendix A.1 (page 135), we list all publications we contributed. In Appendix A.2 (page 137), we collect the corresponding permissions of reuse.

A.1 List of Publications

Our main research concern concerning causal discovery in practice has been as been published at international conferences, journals, and workshops, e.g., KDD, ECML PKDD, NeurIPS, and IJCAI.

- Huegle, J.; Hagedorn, C.; Schlosser, R.: *A kNN-Based Non-Parametric Conditional Independence Test for Mixed Data and Application in Causal Discovery*. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), Part I*. 2023, pp. 541 – 558.
- Hagedorn, C.; Huegle, J.; Schlosser, R.: *Understanding Unforeseen Production Downtimes in Manufacturing Processes Using Log Data-Driven Causal Reasoning*. In *Journal of Intelligent Manufacturing* 33(7), 2022: pp. 2027–2043[11].
- Huegle, J.; Hagedorn, C.; Uflacker, M.: *Unterstützte Fehlerbehebung durch kausales Strukturwissen in Überwachungssystemen der Automobilfertigung*. In *Software Engineering (SE)*. Gesellschaft für Informatik, 2021, pp. 1–2.
- Huegle, J.; Hagedorn, C.; Perscheid, M.; Plattner, H.: *MPCSL - A Modular Pipeline for Causal Structure Learning*. In *Proceedings of the Conference on Knowledge Discovery and Data Mining (KDD)*. 2021, pp. 3068–3076.
- Huegle, J.: *An Information-Theoretic Approach on Causal Structure Learning for Heterogeneous Data Characteristics of Real-World Scenarios*. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Doctoral Consortium Track*. 2021, pp. 4891–4892.
- Huegle, J.; Hagedorn, C.; Böhme, L.; Pörschke, M.; Umland, J.; Schlosser, R.: *MANM-CS: Data Generation for Benchmarking Causal Structure Learning from Mixed Discrete-Continuous and Nonlinear Data*. In *Neural Information Processing Systems (NeurIPS), Workshop on Causal Inference and Machine Learning: Why Now?* 2021: pp. 1–15.
- Huegle, J.; Hagedorn, C.; Uflacker, M.: *How Causal Structural Knowledge Adds Decision Support in Monitoring of Automotive Body Shop Assembly Lines*. In *Proceedings of the International Joint Conference on Artificial Intelligence (ICJAI), Demos*. 2020, pp. 5246–5248.

[11] Note that the first two authors contributed equally to this work.

Our complementary contributions to hardware acceleration for causal discovery have been published at international conferences, workshops, and technical reports.

- Braun, T.; Hurdelhey, B.; Meier, D.; Tsayun, P.; Hagedorn, C.; Huegle, J.; Schlosser, R.: *GPUCSL: GPU-Based Library for Causal Structure Learning*. In *Proceedings of the International Conference on Data Mining (ICDM), Open Project Forum*. 2022, pp. 1236–1239.
- Hagedorn, C.; Lange, C.; Huegle, J.; Schlosser, R.: *GPU Acceleration for Information-Theoretic Constraint-Based Causal Discovery*. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), Workshop on Causal Discovery*. 2022, pp. 30–60.
- Schmidt, C.; Huegle, J.; Horschig, S.; Uflacker, M.: *Strategies for an Improved GPU-Accelerated Skeleton Discovery for Gaussian Distribution Models*. In *Proceedings of the 2018 HPI Future SOC Lab, Technical Reports*. 2022, pp. 187–197.
- Hagedorn, C.; Huegle, J.: *GPU-Accelerated Constraint-Based Causal Structure Learning for Discrete Data*. In *Proceedings of the International Conference on Data Mining (SDM)*. 2021, pp. 37–45.
- Hagedorn, C.; Huegle, J.: *Constraint-Based Causal Structure Learning in Multi-GPU Environments*. In *Proceedings of the Lernen. Wissen. Daten. Analysen. (LWDA), Workshop of Fachgruppe Knowledge Discovery and Machine Learning (KDML)*. 2021, pp. 106–118.
- Schmidt, C.; Huegle, J.: *Towards a GPU-Accelerated Causal Inference*. In *Proceedings of the 2017 HPI Future SOC Lab, Technical Reports*. 2020, pp. 187–194.
- Schmidt, C.; Huegle, J.; Horschig, S.; Uflacker, M.: *Out-of-Core GPU-Accelerated Causal Structure Learning*. In *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*. 2019, pp. 89–104.
- Schmidt, C.; Huegle, J.; Bode, P.; Uflacker, M.: *Load-Balanced Parallel Constraint-Based Causal Structure Learning on Multi-Core Systems for High-Dimensional Data*. In *Proceedings of the Conference on Knowledge Discovery and Data Mining (KDD), Workshop on Causal Discovery*. 2019, pp. 59–77.
- Schmidt, C.; Huegle, J.; Uflacker, M.: *Order-Independent Constraint-Based Causal Structure Learning for Gaussian Distribution Models Using GPUs*. In *Proceedings of the International Conference on Scientific and Statistical Database Management (SSDBM)* 2018: pp. 19:1–19:10.

Furthermore, we published and contributed to research that addresses other domains, such as recommence markets, agile software development, and data stream processing.

- Groeneveld, J.; Herrmann, J.; Mollenhauer, N.; Dreessen, L.; Bessin, N.; Schulze Tast, J.; Kastius, A.; Huegle, J.; Schlosser, R.: *Self-Learning Agents for Recommerce Markets*. In *Business and Information Systems Engineering (BISE)*. 2023. *(To Appear)*.
- Matthies, C.; Huegle, J.; Dürschmid, T.; Teusner, R.: *Attitudes, Beliefs, and Development Data Concerning Agile Software Development Practices*. In *Software Engineering (SE)*. 2020, pp. 73–74.
- Hesse, G.; Matthies, C.; Glass, K.; Huegle, J.; Uflacker, M.: *Quantitative Impact Evaluation of an Abstraction Layer for Data Stream Processing Systems*. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*. 2019, pp. 1381–1392.
- Matthies, C.; Huegle, J.; Dürschmid, T.; Teusner, R.: *Attitudes, Beliefs, and Development Data Concerning Agile Software Development Practices*. In *Proceedings of the International Conference on Software Engineering (ICSE), Software Engineering Education and Training*. 2019, pp. 158–169.

A.2 Permission for Reuse of Published Material

Reuse of Material Published by ACM

Authors can reuse any portion of their own work in a new work of their own (and no fee is expected) as long as a citation and DOI pointer to the Version of Record in the ACM Digital Library are included.
Contributing complete papers to any edited collection of reprints for which the author is not the editor, requires permission and usually a republication fee.
Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included. Authors can use any portion of their own work in presentations and in the classroom (and no fee is expected).

Reuse of Material Published by IEEE

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Hasso Plattner Institute's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to `http://www.ieee.org/publications_standards/publications/rights/right_link.html` to learn how to obtain a License from RightsLink.

Reuse of Material Published by SIAM

SIAM has sole use for distribution in all forms and media, such as microfilm and anthologies, except that the author(s) or, in the case of a "work made for hire" the employer will retain: The right to use all or part of the content of the paper in future works of the author(s), including the author's teaching, technical collaborations, conference presentations, lectures, other scholarly works and professional activities, or any other activity falling under the fair use provisions of the U.S. Copyright Act. If the copyright is granted to SIAM, then proper notice of SIAM's copyright should be provided.

Reuse of Material Published by Springer

Authors have the right to reuse their article's Version of Record, in whole or in part, in their own thesis. Additionally, they may reproduce and make available their thesis, including Springer Nature content, as required by their awarding academic institution. Authors must properly cite the published article in their thesis according to current citation standards.

List of Figures

List of Tables

List of Abbreviations

ACLG	augmented conditional linear Gaussian
ANM	additive noise model
CGM	causal graphical model
CI	conditional independence
CLG	conditional linear Gaussian
CMC	causal Markov condition
CMI	conditional mutual information
CP	conditional permutation
CPDAG	complete partially directed acyclic graph
CPU	central processing unit
CTF	common task framework
DAG	directed acyclic graph
GPU	graphics processing unit
kNN	k-nearest neighbors
MANM	mixed additive noise model
MDL	minimum description length
MI	mutual information
RCA	root cause analysis
ROC AUC	area under the receiver operating curve
SCM	structural causal model
SHD	structural Hamming distance

References

[1] Abellan-Nebot, J. V.; Subirón, F. R.: *A Review of Machining Monitoring Systems Based on Artificial Intelligence Process Models*. In *The International Journal of Advanced Manufacturing Technology* 47(1-4), 2010: pp. 237–257

[2] Andrews, B.; Ramsey, J.; Cooper, G. F.: *Scoring Bayesian Networks of Mixed Variables*. In *International Journal of Data Science and Analytics* 6(1), 2018: pp. 3–18

[3] Antos, A.; Kontoyiannis, I.: *Convergence Properties of Functional Estimates for Discrete Distributions*. In *Random Structures and Algorithms* 19(3-4), 2001: pp. 163–193

[4] Baba, K.; Shibata, R.; Sibuya, M.: *Partial Correlation and Conditional Correlation as Measures of Conditional Independence*. In *Australian and New Zealand Journal of Statistics* 46(4), 2004: pp. 657–664

[5] Banf, M.; Rhee, S. Y.: *Computational Inference of Gene Regulatory Networks: Approaches, Limitations and Opportunities*. In *Biochimica et Biophysica Acta (BBA), Gene Regulatory Mechanisms* 1860(1), 2017: pp. 41–52

[6] Beinlich, I. A.; Suermondt, H. J.; Chavez, R. M.; Cooper, G. F.: *The ALARM Monitoring System: A Case Study with two Probabilistic Inference Techniques for Belief Networks*. In *Proceedings of the European Conference on Artificial Intelligence in Medicin (AIME)*. 1989, pp. 247–256

[7] Berrett, T. B.; Wang, Y.; Barber, R. F.; Samworth, R. J.: *The Conditional Permutation Test for Independence While Controlling for Confounders*. In *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 82(1), 2020: pp. 175–197

[8] Binder, J.; Koller, D.; Russell, S.; Kanazawa, K.: *Adaptive Probabilistic Networks with Hidden Variables*. In *Machine Learning* 29(2), 1997: pp. 213–244

[9] Borboudakis, G.; Tsamardinos, I.: *Incorporating Causal Prior Knowledge as Path-Constraints in Bayesian Networks and Maximal Ancestral Graphs*. In *Proceedings of the International Conference on Machine Learning (ICML)*. 2012, pp. 427–434

[10] Boudjelida, A.: *On the Robustness of Joint Production and Maintenance Scheduling in Presence of Uncertainties*. In *Journal of Intelligent Manufacturing* 30(4), April 2019: pp. 1515–1530

[11] Bradley, J. V.: *Distribution-Free Statistical Tests*. Prentice Hall, Inc., 1968

[12] Braun, T.; Hurdelhey, B.; Meier, D.; Tsayun, P.; Hagedorn, C.; Huegle, J.; Schlosser, R.: *GPUCSL: GPU-Based Library for Causal Structure Learning*. In *Proceedings of the International Conference on Data Mining (ICDM), Open Project Forum*. 2022, pp. 1236–1239

[13] Brouillard, P.; Lachapelle, S.; Lacoste, A.; Lacoste-Julien, S.; Drouin, A.: *Differentiable Causal Discovery from Interventional Data*. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2020, pp. 21865–21877

[14] Cabeli, V.; Verny, L.; Sella, N.; Uguzzoni, G.; Verny, M.; Isambert, H.: *Learning Clinical Networks from Medical Records Based on Information Estimates in Mixed-Type Data*. In *Public Library of Science (PLoS), Computational Biology* 16(5), 2020: pp. 1–19

[15] Chen, K. S.; Huang, M.: *Performance Measurement for a Manufacturing System Based on Quality, Cost and Time*. In *International Journal of Production Research* 44(11), 2006: pp. 2221–2243

[16] Cheng, L.; Guo, R.; Moraffah, R.; Sheth, P.; Candan, K. S.; Liu, H.: *Evaluation Methods and Measures for Causal Learning Algorithms*. In *IEEE Transactions on Artificial Intelligence* 3, 2022: pp. 924–943

[17] Chickering, D. M.: *Optimal Structure Identification with Greedy Search*. In *Journal of Machine Learning Resesarch* 3, March 2003: pp. 507–554

[18] Chien, C.-F.; Chuang, S.-C.: *A Framework for Root Cause Detection of Sub-Batch Processing System for Semiconductor Manufacturing Big Data Analytics*. In *IEEE Transactions on Semiconductor Manufacturing* 27(4), 2014: pp. 475–488

[19] Chryssolouris, G.: *Manufacturing Systems: Theory and Practice*. Springer, 2013

[20] Colombo, D.; Maathuis, M. H.: *Order-Independent Constraint-Based Causal Structure Learning*. In *Journal of Machine Learning Research* 15, 2014: pp. 3921–3962

[21] Colombo, D.; Maathuis, M. H.; Kalisch, M.; Richardson, T. S.: *Learning High-Dimensional Directed Acyclic Graphs with Latent and Selection Variables*. In *The Annals of Statistics* 40, 2012: pp. 294–321

[22] Conati, C.; Gertner, A. S.; VanLehn, K.; Druzdzel, M. J.: *On-Line Student Modeling for Coached Problem Solving Using Bayesian Networks*. In *Proceedings of the International Conference on User Modeling, Adaptation and Personalization (UMAP)*. 1997, pp. 231–242

[23] Conrady, S.; Jouffe, L.: *Bayesian Networks and BayesiaLab: A Practical Introduction for Researchers*. Bayesia, 2015

[24] Cui, R.; Groot, P.; Heskes, T.: *Copula PC Algorithm for Causal Discovery from Mixed Data*. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*. 2016, pp. 377–392

[25] Cui, R.; Groot, P.; Schauer, M.; Heskes, T.: *Learning the Causal Structure of Copula Models with Latent Variables*. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. 2018, pp. 188–197

[26] Dagum, L.; Menon, R.: *OpenMP: An Industry-Standard API for Shared-Memory Programming*. In *IEEE Computational Science and Engineering* 5(1), January 1998: pp. 46–55

[27] Davis, J.; Edgar, T.; Graybill, R.; Korambath, P.; Schott, B.; Swink, D.; Wang, J.; Wetzel, J.: *Smart Manufacturing*. In *Annual Review of Chemical and Biomolecular Engineering* 6(1), 2015: pp. 141–160

[28] Dawid, A. P.: *Conditional Independence in Statistical Theory*. In *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 41(1), 1979: pp. 1–31

[29] Deckert, A. C.; Kummerfeld, E.: *Investigating the Effect of Binning on Causal Discovery*. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* 2019: pp. 2574–2581

[30] Donoho, D.: *50 Years of Data Science*. In *Journal of Computational and Graphical Statistics* 26(4), 2017: pp. 745–766

[31] Dorie, V.; Hill, J.; Shalit, U.; Scott, M.; Cervone, D.: *Automated versus Do-It-Yourself Methods for Causal Inference: Lessons Learned from a Data Analysis Competition*. In *Statistical Science* 34(1), 2019: pp. 43–68

[32] Dougherty, J.; Kohavi, R.; Sahami, M.: *Supervised and Unsupervised Discretization of Continuous Features*. In *Proceedings of the International Conference on Machine Learning (ICML)*. 1995, pp. 194–202

[33] Du, S.; Lv, J.; Xi, L.: *A Robust Approach for Root Causes Identification in Machining Processes using Hybrid Learning Algorithm and Engineering Knowledge*. In *Journal of Intelligent Manufacturing* 23(5), 2012: pp. 1833–1847

[34] e Oliveira, E.; Miguéis, V. L.; Borges, J.: *Understanding Overlap in Automatic Root Cause Analysis in Manufacturing Using Causal Inference*. In *IEEE Access* 10, 12 2021: pp. 191–201

[35] e Oliveira, E.; Miguéis, V. L.; Borges, J.: *Automatic Root Cause Analysis in Manufacturing: An Overview and Conceptualization*. In *Journal of Intelligent Manufacturing* 34(5), 2022: pp. 2061–2078

[36] Edwards, D.: *Introduction to Graphical Modelling*. Springer, 2012

[37] Emmert-Streib, F.; Glazko, G.; De Matos Simoes, R.; et al.: *Statistical Inference and Reverse Engineering of Gene Regulatory Networks from Observational Expression Data*. In *Frontiers in Genetics* 3, 2012: pp. 1–8

[38] Ernst, M. D.: *Permutation Methods: A Basis for Exact Inference*. In *Statistical Science* 19(4), 2004: pp. 676–685

[39] Foraita, R.; Friemel, J.; Günther, K.; Behrens, T.; Bullerdiek, J.; Nimzyk, R.; Ahrens, W.; Didelez, V.: *Causal Discovery of Gene Regulation with Incomplete Data*. In *Journal of the Royal Statistical Society Series A: Statistics in Society* 183(4), 2020: pp. 1747–1775

[40] Forré, P.; Mooij, J. M.: *Causal Calculus in the Presence of Cycles, Latent Confounders and Selection Bias*. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. 2020, pp. 71–80

[41] Frenzel, S.; Pompe, B.: *Partial Mutual Information for Coupling Analysis of Multivariate Time Series*. In *Physical Review Letters* 99(20), 2007: pp. 1–4

[42] Frigyesi, A.; Hössjer, O.: *A Test for Singularity*. In *Statistics and probability letters* 40(3), 1998: pp. 215–226

[43] Gao, W.; Kannan, S.; Oh, S.; Viswanath, P.: *Estimating Mutual Information for Discrete-Continuous Mixtures*. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*. 2017, pp. 5988–5999

[44] Glymour, C.; Zhang, K.; Spirtes, P.: *Review of Causal Discovery Methods Based on Graphical Models*. In *Frontiers in Genetics* 10, 2019: 524

[45] Gray, R. M.: *Entropy and Information Theory*. Springer, 2011

[46] Greenfield, A.; Madar, A.; Ostrer, H.; Bonneau, R.: *DREAM4: Combining Genetic and Dynamic Information to Identify Biological Networks and Dynamical Models*. In *Public Library of Science (PLoS), ONE)* 5(10), 2010: pp. 1–14

[47] Grinberg, M.: *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, Inc., 1. Edition, 2014

[48] Groeneveld, J.; Herrmann, J.; Mollenhauer, N.; Dreessen, L.; Bessin, N.; Schulze Tast, J.; Kastius, A.; Huegle, J.; Schlosser, R.: *Self-Learning Agents for Recommerce Markets*. In *Business and Information Systems Engineering (BISE)*. 2023. *(To Appear)*

[49] Guo, R.; Cheng, L.; Li, J.; Hahn, P. R.; Liu, H.: *A Survey of Learning Causality with Data: Problems and Methods*. In *ACM Computing Surveys* 53(4), 2020: pp. 75:1–75:37

[50] Gutschi, C.; Furian, N.; Suschnigg, J.; Neubacher, D.; Voessner, S.: *Log-Based Predictive Maintenance in Discrete Parts Manufacturing*. In *Proceedings of the Conference on Intelligent Computation in Manufacturing Engineering (CIRP)* 79, 2019: pp. 528–533

[51] Hagedorn, C.: *Parallel Execution of Causal Structure Learning on Graphics Processing Units*. Universität Potsdam, 2023. Doctoral Thesis

[52] Hagedorn, C.; Huegle, J.: *Constraint-Based Causal Structure Learning in Multi-GPU Environments*. In *Proceedings of the Lernen. Wissen. Daten. Analysen. (LWDA), Workshop of Fachgruppe Knowledge Discovery and Machine Learning (KDML)*. 2021, pp. 106–118

[53] Hagedorn, C.; Huegle, J.: *GPU-Accelerated Constraint-Based Causal Structure Learning for Discrete Data*. In *Proceedings of the International Conference on Data Mining (SDM)*. 2021, pp. 37–45

[54] Hagedorn, C.; Huegle, J.; Schlosser, R.: *Understanding Unforeseen Production Downtimes in Manufacturing Processes Using Log Data-Driven Causal Reasoning*. In *Journal of Intelligent Manufacturing* 33(7), 2022: pp. 2027–2043

[55] Hagedorn, C.; Lange, C.; Huegle, J.; Schlosser, R.: *GPU Acceleration for Information-Theoretic Constraint-Based Causal Discovery*. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), Workshop on Causal Discovery*. 2022, pp. 30–60

[56] Hauser, A.; Bühlmann, P.: *Characterization and Greedy Learning of Interventional Markov Equivalence Classes of Directed Acyclic Graphs*. In *Journal of Machine Learning Research* 13(1), 2012: p. 2409–2464

[57] Heinze-Deml, C.; Maathuis, M. H.; Meinshausen, N.: *Causal Structure Learning*. In *Annual Review of Statistics and Its Application* 5(1), 2018: pp. 371–391

[58] Hernán, M. A.; et al.: *Comment: Spherical Cows in a Vacuum: Data Analysis Competitions for Causal Inference*. In *Statistical Science* 34(1), 2019: pp. 69–71

[59] Hernán, M. A.; Robins, J. M.: *Causal Inference: What If*. CRC Press, 2023

[60] Hesse, G.; Matthies, C.; Glass, K.; Huegle, J.; Uflacker, M.: *Quantitative Impact Evaluation of an Abstraction Layer for Data Stream Processing Systems*. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*. 2019, pp. 1381–1392

[61] Higgins, J. J.: *An Introduction to Modern Nonparametric Statistics*. Brooks/Cole, 2004

[62] Holland, P. W.: *Statistics and Causal Inference*. In *Journal of the American Statistical Association* 81(396), 1986: pp. 945–960

[63] Hoyer, P. O.; Janzing, D.; Mooij, J.; Peters, J.; Schölkopf, B.: *Nonlinear Causal Discovery with Additive Noise Models*. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*. 2008, pp. 689–696

[64] Huang, B.; Zhang, K.; Lin, Y.; Schölkopf, B.; Glymour, C.: *Generalized Score Functions for Causal Discovery*. In *Proceedings of the International Conference on Knowledge Discovery; Data Mining (KDD)*. 2018, pp. 1551–1560

[65] Huang, T.-M.: *Testing Conditional Independence Using Maximal Nonlinear Conditional Correlation*. In *The Annals of Statistics* 38(4), 2010: pp. 2047–2091

[66] Huegle, J.: *An Information-Theoretic Approach on Causal Structure Learning for Heterogeneous Data Characteristics of Real-World Scenarios*. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Doctoral Consortium Track*. 2021, pp. 4891–4892

[67] Huegle, J.; Hagedorn, C.; Böhme, L.; Pörschke, M.; Umland, J.; Schlosser, R.: *MANM-CS: Data Generation for Benchmarking Causal Structure Learning from Mixed Discrete-Continuous and Nonlinear Data*. In *Neural Information Processing Systems (NeurIPS), Workshop on Causal Inference and Machine Learning: Why Now?* 2021: pp. 1–15

[68] Huegle, J.; Hagedorn, C.; Perscheid, M.; Plattner, H.: *MPCSL - A Modular Pipeline for Causal Structure Learning*. In *Proceedings of the Conference on Knowledge Discovery and Data Mining (KDD)*. 2021, pp. 3068–3076

[69] Huegle, J.; Hagedorn, C.; Schlosser, R.: *A kNN-Based Non-Parametric Conditional Independence Test for Mixed Data and Application in Causal Discovery*. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), Part I*. 2023, pp. 541 – 558

[70] Huegle, J.; Hagedorn, C.; Uflacker, M.: *How Causal Structural Knowledge Adds Decision Support in Monitoring of Automotive Body Shop Assembly Lines*. In *Proceedings of the International Joint Conference on Artificial Intelligence (IC-JAI), Demos*. 2020, pp. 5246–5248

[71] Huegle, J.; Hagedorn, C.; Uflacker, M.: *Unterstützte Fehlerbehebung durch kausales Strukturwissen in Überwachungssystemen der Automobilfertigung*. In *Software Engineering (SE)*. Gesellschaft für Informatik, 2021, pp. 1–2

[72] Janzing, D.; Balduzzi, D.; Grosse-Wentrup, M.; Schölkopf, B.: *Quantifying Causal Influences*. In *The Annals of Statistics* 41(5), 2013: pp. 2324 – 2358

[73] Jin, R.; Breitbart, Y.; Muoh, C.: *Data Discretization Unification*. In *Proceedings of the International Conference on Data Mining (ICDM)*. 2007, pp. 183–192

[74] Kalisch, M.; Bühlmann, P.: *Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm*. In *Journal of Machine Learning Research* 8, 2007: pp. 613–636

[75] Kalisch, M.; Mächler, M.; Colombo, D.; Maathuis, M. H.; Bühlmann, P.: *Causal Inference Using Graphical Models with the R Package pcalg*. In *Journal of Statistical Software* 47(11), 2012: pp. 1–26

[76] Karavani, E.; El-Hay, T.; Shimoni, Y.; Yanover, C.: *Comment: Causal Inference Competitions: Where Should We Aim?*. In *Statistical Science* 34(1), 2019: pp. 86–89

[77] Khatab, A.: *Maintenance Optimization in Failure-Prone Systems under Imperfect Preventive Maintenance*. In *Journal of Intelligent Manufacturing* 29(3), 2018: pp. 707–717

[78] Kim, I.; Neykov, M.; Balakrishnan, S.; Wasserman, L.: *Local Permutation Tests for Conditional Independence*. In *The Annals of Statistics* 50(6), 2022: pp. 3388–3414

[79] Kiriakidou, N.; Diou, C.: *An Evaluation Framework for Comparing Causal Inference Models*. In *Proceedings of the Hellenic Conference on Artificial Intelligence (SETN)* 2022: pp. 34:1 – 34:9

[80] Korb, K. B.; Nicholson, A. E.: *Bayesian Artificial Intelligence*. CRC Press, Inc., 2. Edition, 2010

[81] Koren, Y.; Hu, S. J.; Weber, T. W.: *Impact of Manufacturing System Configuration on Performance*. In *CIRP Annals* 47(1), 1998: pp. 369–372

[82] Kozachenko, L. F.; Leonenko, N. N.: *Sample Estimate of the Entropy of a Random Vector*. In *Problems of Information Transmission* 23(2), 1987: pp. 9–16

[83] Kraskov, A.; Stögbauer, H.; Grassberger, P.: *Estimating Mutual Information*. In *Physical Review E* 69(6), 2004: 066138

[84] Kühnert, C.; Beyerer, J.: *Data-Driven Methods for the Detection of Causal Structures in Process Technology*. In *Machines* 2(4), 2014: pp. 255–274

[85] Kummerfeld, E.; Rix, A.: *Simulations Evaluating Resampling Methods for Causal Discovery: Ensemble Performance and Calibration*. In *Preceedings of the International Conference on Bioinformatics and Biomedicine (BIBM)*. 2019, pp. 2586–2593

[86] Lagani, V.; Athineou, G.; Farcomeni, A.; Tsagris, M.; Tsamardinos, I.: *Feature Selection with the R Package MXM: Discovering Statistically Equivalent Feature Subsets*. In *Journal of Statistical Software* 80(7), 2017: pp. 1–25

[87] Lauritzen, S. L.; Spiegelhalter, D. 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. 157–224

[88] Lawrence, A. R.; Kaiser, M.; Sampaio, R.; Sipos, M.: *Data Generating Process to Evaluate Causal Discovery Techniques for Time Series Data*. In *Neural Information Processing Systems (NeurIPS), Workshop on Causal Discovery and Causality-Inspired Machine Learning* 2020: pp. 1–17

[89] Le, T. D.; Hoang, T.; Li, J.; Liu, L.; Liu, H.; Hu, S.: *A Fast PC Algorithm for High Dimensional Causal Discovery with Multi-Core PCs*. In *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16(5), 2019: pp. 1483–1495

[90] Le, T. D.; Xu, T.; Liu, L.; Shu, H.; Hoang, T.; Li, J.: *ParallelPC: An R Package for Efficient Causal Exploration in Genomic Data*. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. 2018, pp. 207–218

[91] Lee, J. D.; Hastie, T. J.: *Learning the Structure of Mixed Graphical Models*. In *Journal of Computational and Graphical Statistics* 24(1), 2015: pp. 230–253

[92] Lehmann, E. L.; D'Abrera, H. J. M.: *Nonparametrics: Statistical Methods Based on Ranks*. Springer, 1975

[93] Lerner, U.; Segal, E.; Koller, D.: *Exact Inference in Networks with Discrete Children of Continuous Parents*. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. 2001, pp. 319—-328

[94] Li, C.; Fan, X.: *On Nonparametric Conditional Independence Tests for Continuous Variables*. In *Wiley Interdisciplinary Reviews: Computational Statistics* 12(3), 2020

[95] Li, J.; Shi, J.: *Knowledge Discovery from Observational Data for Process Control Using Causal Bayesian Networks*. In *Institute of Industrial Engineers Transactions* 39(6), 2007: pp. 681–690

[96] Li, Z.; Wang, Y.; Wang, K.: *A Data-Driven Method Based on Deep Belief Networks for Backlash Error Prediction in Machining Centers*. In *Journal of Intelligent Manufacturing* 31(7), 2020: pp. 1693–1705

[97] Liang, S. Y.; Hecker, R. L.; Landers, R. G.: *Machining Process Monitoring and Control: The State-of-the-Art*. In *Journal of Manufacturing Science and Engineering* 126(2), 2004: pp. 297–310

[98] Lin, A.; Merchant, A.; Sarkar, S. K.; D'Amour, A.: *Universal Causal Evaluation Engine: An API for Empirically Evaluating Causal Inference Models*. In *Proceedings of the Conference on Knowledge Discovery and Data Mining (KDD), Workshop on Causal Discovery*. 2019, pp. 50–58

[99] Liu, J.; Chang, Q.; Xiao, G.; Biller, S.: *The Costs of Downtime Incidents in Serial Multistage Manufacturing Systems*. In *Journal of Manufacturing Science and Engineering* 134(2), 2012: pp. 1–10

[100] Liu, L.; Mishchenko, M. I.; Patrick Arnott, W.: *A Study of Radiative Properties of Fractal Soot Aggregates Using the Superposition T-Matrix Method*. In *Journal of Quantitative Spectroscopy and Radiative Transfer* 109(15), 2008: pp. 2656–2663

[101] Liu, Q.; Dong, M.; Lv, W.; Ye, C.: *Manufacturing System Maintenance Based on Dynamic Programming Model with Prognostics Information*. In *Journal of Intelligent Manufacturing* 30(3), March 2019: pp. 1155–1173

[102] Liu, T.; Yuan, R.; Chang, H.: *Research on the Internet of Things in the Automotive Industry*. In *Proceedings of the International Conference on Management of e-Commerce and e-Government (ICMECG)*. 2012, pp. 230–233

[103] Lloyd, S.: *Least Squares Quantization in PCM*. In *IEEE Transactions on Information Theory* 28(2), 1982: pp. 129–137

[104] Maathuis, M.; Drton, M.; Lauritzen, S.; Wainwright, M.: *Handbook of Graphical Models*. CRC Press, Inc., 1. Edition, 2018

[105] Magrini, A.; Blasi, S. D.; Stefanini, F. M.: *A Conditional Linear Gaussian Network to Assess the Impact of Several Agronomic Settings on the Quality of Tuscan Sangiovese Grapes*. In *Biometrical Letters* 54(1), 2017: pp. 25–42

[106] Malinsky, D.; Danks, D.: *Causal Discovery Algorithms: A Practical Guide*. In *Philosophy Compass* 13(1), 2018: pp. 1–11

[107] Mandros, P.; Kaltenpoth, D.; Boley, M.; Vreeken, J.: *Discovering Functional Dependencies from Mixed-Type Data*. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*. 2020, pp. 1404–1414

[108] Marazopoulou, K.; Ghosh, R.; Lade, P.; Jensen, D. D.: *Causal Discovery for Manufacturing Domains*. In *Computing Research Repository (CoRR)* abs/1605.04056, 2016

[109] Marbach, D.; ; Costello, J. C.; Küffner, R.; Vega, N. M.; Prill, R. J.; Camacho, D. M.; Allison, K. R.; Kellis, M.; Collins, J. J.; Stolovitzky, G.: *Wisdom of Crowds for Robust Gene Network Inference*. In *Nature Methods* 9(8), 8 2012: pp. 796–804

[110] Margaritis, D.: *Distribution-Free Learning of Bayesian Network Structure in Continuous Domains*. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. 2005, p. 825–830

[111] Marx, A.; Gretton, A.; Mooij, J. M.: *A Weaker Faithfulness Assumption based on Triple Interactions*. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. 2021, pp. 451–460

[112] Marx, A.; Yang, L.; van Leeuwen, M.: *Estimating Conditional Mutual Information for Discrete-Continuous Mixtures using Multi-Dimensional Adaptive Histograms*. In *Proceedings of the International Conference on Data Mining (SDM)*. 2021, pp. 387–395

[113] Matthies, C.; Huegle, J.; Dürschmid, T.; Teusner, R.: *Attitudes, Beliefs, and Development Data Concerning Agile Software Development Practices*. In *Proceedings of the International Conference on Software Engineering (ICSE), Software Engineering Education and Training*. 2019, pp. 158–169

[114] Matthies, C.; Huegle, J.; Dürschmid, T.; Teusner, R.: *Attitudes, Beliefs, and Development Data Concerning Agile Software Development Practices*. In *Software Engineering (SE)*. 2020, pp. 73–74

[115] Meek, C.: *Causal Inference and Causal Explanation with Background Knowledge*. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. 1995, pp. 403–410

[116] Merkel, D.: *Docker: Lightweight Linux Containers for Consistent Development and Deployment*. In *Linux Journal* 2014(239), 2014

[117] Mesner, O. C.; Shalizi, C. R.: *Conditional Mutual Information Estimation for Mixed, Discrete and Continuous Data*. In *IEEE Transactions on Information Theory* 67(1), 2021: pp. 464–484

[118] Mobley, R. K.: *An Introduction to Predictive Maintenance*. Butterworth-Heinemann, 2. Edition, 2002

[119] Montgomery, D. C.: *Introduction to Statistical Quality Control*. John Wiley & Sons, Inc., 2007

[120] Mooij, J. M.; Peters, J.; Janzing, D.; Zscheischler, J.; Schölkopf, B.: *Distinguishing Cause from Effect Using Observational Data: Methods and Benchmarks*. In *Journal of Machine Learning Research* 17(1), 2016: p. 1103–1204

[121] Neapolitan, R. E.: *Learning Bayesian Networks*. Prentice Hall, Inc., 2003

[122] Nickolls, J.; Buck, I.; Garland, M.; Skadron, K.: *Scalable Parallel Programming with CUDA*. In *Queue* 6(2), March 2008: p. 40–53

[123] Nikula, R.-P.; Karioja, K.; Leiviskä, K.; Juuso, E.: *Prediction of Mechanical Stress in Roller Leveler Based on Vibration Measurements and Steel Strip Properties*. In *Journal of Intelligent Manufacturing* 30(4), 2019: pp. 1563–1579

[124] Obe, R.; Hsu, L.: *PostgreSQL: Up and Running*. O'Reilly Media, Inc., 2012

[125] Opgen-Rhein, R.; Strimmer, K.: *From Correlation to Causation Networks: A Simple Approximate Learning Algorithm and its Application to High-Dimensional Plant Gene Expression Data*. In *BMC Systems Biology* 1(1), 2007: pp. 1–10

[126] Parnas, D. L.: *On the Criteria to Be Used in Decomposing Systems into Modules*. In *Communications of the ACM* 15(12), 1972: p. 1053–1058

[127] Pearl, J.: *Comment: Graphical Models, Causality and Intervention*. In *Statistical Science* 8(3), 1993: pp. 266–269

[128] Pearl, J.: *Causal Diagrams for Empirical Research*. In *Biometrika* 82(4), 1995: pp. 669–688

[129] Pearl, J.: *Causal Inference in Statistics: An Overview*. In *Statistics Surveys* 3, 2009: pp. 96–146

[130] Pearl, J.: *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2. Edition, 2009

[131] Pearl, J.; Mackenzie, D.: *The Book of Why: The New Science of Cause and Effect*. Basic Books, 2018

[132] Pearson, K. F.: *X. On the Criterion that a Given System of Deviations from the Probable in the Case of a Correlated System of Variables is such that it can be Reasonably Supposed to have Arisen from Random Sampling*. In *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50(302), 1900: pp. 157–175

[133] Peters, J.; Bühlmann, P.; Meinshausen, N.: *Causal Inference by Using Invariant Prediction: Identification and Confidence Intervals*. In *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 78(5), 2016: pp. 947–1012

[134] Peters, J.; Janzing, D.; Schölkopf, B.: *Causal Inference on Discrete Data using Additive Noise Models*. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(12), 2011: pp. 2436–2450

[135] Peters, J.; Janzing, D.; Schölkopf, B.: *Elements of Causal Inference - Foundations and Learning Algorithms*. Adaptive Computation and Machine Learning Series, MIT Press, 2017

[136] Peters, J.; Janzing, D.; Schölkopf, B.: *Identifying Cause and Effect on Discrete Data using Additive Noise Models*. In *Proceedings of the Conference on Artificial Intelligence and Statistics (AISTATS)*. 2010, pp. 597–604

[137] Phipson, B.; Smyth, G. K.: *Permutation P-Values Should Never Be Zero: Calculating Exact P-Values When Permutations Are Randomly Drawn*. In *Statistical Applications in Genetics and Molecular Biology* 9(1), 2010: pp. 39:1–39:12

[138] Qin, W.; Zha, D.; Zhang, J.: *An Effective Approach for Causal Variables Analysis in Diesel Engine Production by using Mutual Information and Network Deconvolution*. In *Journal of Intelligent Manufacturing* 31, 2020: pp. 1661–1671

[139] Raghu, V. K.; Poon, A.; Benos, P. V.: *Evaluation of Causal Structure Learning Methods on Mixed Data Types*. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), Workshop on Causal Discovery*. 2018, pp. 48–65

[140] Ramm, M.; Bayer, M.; Rhodes, B.: *SQLAlchemy: Database Access Using Python*. Addison Wesley Professional, 2011

[141] Rau, A.; Jaffrézic, F.; Nuel, G.: *Joint Estimation of Causal Effects from Observational and Intervention Gene Expression Data*. In *BMC Systems Biology* 7(1), 2013: pp. 111:1–111:12

[142] Reichenbach, H.: *The Direction of Time*. Dover Publications, Mineola, N.Y., 1956

[143] Reisach, A.; Seiler, C.; Weichwald, S.: *Beware of the Simulated DAG! Causal Discovery Benchmarks May Be Easy to Game*. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2021, pp. 27772–27784

[144] Rodríguez, A. R.; Bernal de Lázaro, J. M.; Prieto-Moreno, A.; da Silva Neto, A. J.; Llanes-Santiago, O.: *An Approach to Robust Fault Diagnosis in Mechanical Systems using Computational Intelligence*. In *Journal of Intelligent Manufacturing* 30(4), 2019: pp. 1601–1615

[145] Rohekar, R. Y.; Nisimov, S.; Gurwicz, Y.; Novik, G.: *Iterative Causal Discovery in the Possible Presence of Latent Confounders and Selection Bias*. In *Advances in Neural Information Processing Systems (NeurIPS)* 2021: pp. 2454–2465

[146] Rokach, L.; Hutter, D.: *Automatic Discovery of the Root Causes for Quality Drift in High Dimensionality Manufacturing Processes*. In *Journal of Intelligent Manufacturing* 23, 2012: pp. 1915–1930

[147] Rothenhäusler, D.; Heinze, C.; Peters, J.; Meinshausen, N.: *BACKSHIFT: Learning Causal Cyclic Graphs from Unknown Shift Interventions*. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2016, pp. 1513–1521

[148] Runge, J.: *Conditional Independence Testing based on a Nearest-Neighbor Estimator of Conditional Mutual Information*. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics (IJCAI)*. 2018, pp. 938–947

[149] Sachs, K.; Perez, O.; Pe'er, D.; Lauffenburger, D. A.; Nolan, G. P.: *Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data*. In *Science* 308(5721), 2005: pp. 523–529

[150] Schäfer, J.; Strimmer, K.: *A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics*. In *Statistical Applications in Genetics and Molecular Biology* 4(1), 2005

[151] Scheines, R.; Spirtes, P.; Glymour, C.; Meek, C.; Richardson, T.: *The TETRAD Project: Constraint Based Aids to Causal Model Specification*. In *Multivariate Behavioral Research* 33(1), 1998: pp. 65–117

[152] Schmidt, C.; Huegle, J.: *Towards a GPU-Accelerated Causal Inference*. In *Proceedings of the 2017 HPI Future SOC Lab, Technical Reports*. 2020, pp. 187–194

[153] Schmidt, C.; Huegle, J.; Bode, P.; Uflacker, M.: *Load-Balanced Parallel Constraint-Based Causal Structure Learning on Multi-Core Systems for High-Dimensional Data*. In *Proceedings of the Conference on Knowledge Discovery and Data Mining (KDD), Workshop on Causal Discovery*. 2019, pp. 59–77

[154] Schmidt, C.; Huegle, J.; Horschig, S.; Uflacker, M.: *Out-of-Core GPU-Accelerated Causal Structure Learning*. In *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*. 2019, pp. 89–104

[155] Schmidt, C.; Huegle, J.; Horschig, S.; Uflacker, M.: *Strategies for an Improved GPU-Accelerated Skeleton Discovery for Gaussian Distribution Models*. In *Proceedings of the 2018 HPI Future SOC Lab, Technical Reports*. 2022, pp. 187–197

[156] Schmidt, C.; Huegle, J.; Uflacker, M.: *Order-Independent Constraint-Based Causal Structure Learning for Gaussian Distribution Models Using GPUs*. In *Proceedings of the International Conference on Scientific and Statistical Database Management (SSDBM)* 2018: pp. 19:1–19:10

[157] Scutari, M.: *Learning Bayesian Networks with the bnlearn R Package*. In *Journal of Statistical Software, Articles* 35(3), 2010: pp. 1–22

[158] Scutari, M.: *Bayesian Network Constraint-Based Structure Learning Algorithms: Parallel and Optimized Implementations in the bnlearn R Package*. In *Journal of Statistical Software* 77(2), 2017: pp. 1–20

[159] Scutari, M.; Graafland, C. E.; Gutiérrez, J. M.: *Who Learns Better Bayesian Network Structures: Constraint-Based, Score-Based or Hybrid Algo-*

rithms?. In *Proceedings of the Ninth International Conference on Probabilistic Graphical Models (PGM)*. 2018, pp. 416–427

[160] Scutari, M.; Howell, P.; Balding, D. J.; Mackay, I.: *Multiple Quantitative Trait Analysis Using Bayesian Networks*. In *Genetics* 198(1), 2014: pp. 129–137

[161] Sedgewick, A. J.; Shi, I.; Donovan, R. M.; Benos, P. V.: *Learning Mixed Graphical Models with Separate Sparsity Parameters and Stability-Based Model Selection*. In *BMC Bioinformatics* 17(5), 2016: pp. 307–318

[162] Shah, R. D.; Peters, J.: *The Hardness of Conditional Independence Testing and the Generalised Covariance Measure*. In *The Annals of Statistics* 48(3), 2020: pp. 1514–1538

[163] Shimizu, S.; Hoyer, P. O.; Hyvärinen, A.; Kerminen, A.: *A Linear Non-Gaussian Acyclic Model for Causal Discovery*. In *Journal of Machine Learning Research* 7(72), 2006: p. 2003–2030

[164] Simpson, E. H.: *The Interpretation of Interaction in Contingency Tables*. In *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 13(2), 1951: pp. 238–241

[165] Singh, K.; Gupta, G.; Tewari, V.; Shroff, G.: *Comparative Benchmarking of Causal Discovery Algorithms*. In *Proceedings of the India Joint International Conference on Data Science and Management of Data (CoDS-COMAD)*. 2018, pp. 46–56

[166] Sipos, R.; Fradkin, D.; Moerchen, F.; Wang, Z.: *Log-Based Predictive Maintenance*. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*. 2014, pp. 1867–1876

[167] Spirtes, P.: *Introduction to Causal Inference*. In *Journal of Machine Learning Research* 11, 2010: pp. 1643–1662

[168] Spirtes, P.; Glymour, C.; Scheines, R.: *Causation, Prediction, and Search*. The MIT Press, 2001

[169] Spirtes, P.; Glymour, C. N.: *An Algorithm for Fast Recovery of Sparse Causal Graphs*. In *Social Science Computer Review* 9(1), 1991: pp. 62–72

[170] Spirtes, P.; Meek, C.; Richardson, T.: *Causal Inference in the Presence of Latent Variables and Selection Bias*. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. 1995, pp. 499–506

[171] Spirtes, P.; Scheines, R.: *Causal Inference of Ambiguous Manipulations*. In *Philosophy of Science* 71(5), 2004: pp. 833–845

[172] Spirtes, P.; Zhang, K.: *Causal Discovery and Inference: Concepts and Recent Methodological Advances*. In *Applied Informatics* 3, 2016: pp. 3:1–3:28

[173] Spirtes, P.; Zhang, K.: *Search for Causal Models*. In *Handbook of Graphical Models*, CRC Press. 2018, pp. 439–470

[174] Strobl, E. V.: *A Constraint-Based Algorithm For Causal Discovery with Cycles, Latent Variables and Selection Bias*. In *International Journal of Data Science and Analytics* 8, 2019: pp. 33–56

[175] Sun, Y.; Qin, W.; Zhuang, Z.; Xu, H.: *An Adaptive Fault Detection and Root-cause Analysis Scheme for Complex Industrial Processes using Moving Window KPCA and Information Geometric Causal Inference*. In *Journal of Intelligent Manufacturing* 32(7), 2021: pp. 2007–2021

[176] Takata, S.; Kirnura, F.; van Houten, F. J.; Westkamper, E.; Shpitalni, M.; Ceglarek, D.; Lee, J.: *Maintenance: Changing Role in Life Cycle Management*. In *CIRP Annals* 53(2), 2004: pp. 643–655

[177] Tikka, S.; Karvanen, J.: *Identifying Causal Effects with the R Package causaleffect*. In *Journal of Statistical Software, Articles* 76(12), 2017: pp. 1–30

[178] Tillman, R. E.; Gretton, A.; Spirtes, P.: *Nonlinear Directed Acyclic Structure Learning with Weakly Additive Noise Models*. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2009, pp. 1847–1855

[179] Tsagris, M.: *Bayesian Network Learning with the PC Algorithm: An Improved and Correct Variation*. In *Applied Artificial Intelligence* 33(2), 2019: pp. 101–123

[180] Tsagris, M.; Borboudakis, G.; Lagani, V.; Tsamardinos, I.: *Constraint-Based Causal Discovery with Mixed Data*. In *International Journal of Data Science and Analytics* 6, 2018: pp. 19–30

[181] Tsamardinos, I.; Borboudakis, G.: *Permutation Testing Improves Bayesian Network Learning*. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), Part III*. 2010, pp. 322–337

[182] Tsamardinos, I.; Brown, L. E.; Aliferis, C. F.: *The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm*. In *Machine Learning* 65, 2006: pp. 31–78

[183] Vitolo, C.; Scutari, M.; Ghalaieny, M.; Tucker, A.; Russell, A.: *Modeling Air Pollution, Climate, and Health Data Using Bayesian Networks: A Case Study of the English Regions*. In *Earth and Space Science* 5(4), 2018: pp. 76–88

[184] Wang, H.; Song, M.: *Ckmeans.1d.dp: Optimal k-Means Clustering in One Dimension by Dynamic Programming*. In *The R Journal* 3(2), 2011: pp. 29–33

[185] Wang, J.; Li, C.; Han, S.; Sarkar, S.; Zhou, X.: *Predictive Maintenance based on Event-Log Analysis: A Case Study*. In *IBM Journal of Research and Development* 61(1), 2017: pp. 11:121–11:132

[186] Woodward, J.: *The Problem of Variable Choice*. In *Synthese* 193(4), 2016: pp. 1047–1072

[187] Wuest, T.; Irgens, C.; Thoben, K.-D.: *An Approach to Monitoring Quality in Manufacturing Using Supervised Machine Learning on Product State Data*. In *Journal of Intelligent Manufacturing* 25(5), 2014: pp. 1167–1180

[188] Wuest, T.; Weimer, D.; Irgens, C.; Thoben, K. D.: *Machine Learning in Manufacturing: Advantages, Challenges, and Applications*. In *Production and Manufacturing Research* 4(1), 2016: pp. 23–45

[189] Ye, N.: *A Reverse Engineering Algorithm for Mining a Causal System Model from System Data*. In *International Journal of Production Research* 55(3), 2017: pp. 828–844

[190] Yu, K.; Guo, X.; Liu, L.; Li, J.; Wang, H.; Ling, Z.; Wu, X.: *Causality-Based Feature Selection: Methods and Evaluations*. In *ACM Computing Surveys* 53(5), 2020: pp. 111:1 – 111:36

[191] Zan, L.; Meynaoui, A.; Assaad, C. K.; Devijver, E.; Gaussier, E.: *A Conditional Mutual Information Estimator for Mixed Data and an Associated Conditional Independence Test*. In *Entropy* 24(9), 2022: 1234

[192] Zarebavani, B.; Jafarinejad, F.; Hashemi, M.; Salehkaleybar, S.: *cuPC: CUDA-based Parallel PC Algorithm for Causal Structure Learning on GPU*. In *IEEE Transactions on Parallel and Distributed Systems* 31(03), 2019: pp. 530–542

[193] Zhang, K.; Hyvärinen, A.: *Causality Discovery with Additive Disturbances: An Information-Theoretical Perspective*. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*. 2009, pp. 570–585

[194] Zhang, K.; Hyvärinen, A.: *On the Identifiability of the Post-Nonlinear Causal Model*. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. 2009, pp. 647–655

[195] Zhang, K.; Peters, J.; Janzing, D.; Schölkopf, B.: *Kernel-Based Conditional Independence Test and Application in Causal Discovery*. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. 2011, pp. 804–813

[196] Zhang, K.; Schölkopf, B.; Spirtes, P.; Glymour, C.: *Learning Causality and Causality-Related Learning: Some Recent Progress*. In *National Science Review* 5(1), 2017: pp. 26–29

[197] Zhang, P.: *Advanced Industrial Control Technology*. William Andrew Publishing, Oxford, 2010

[198] Zhao, P.; Lai, L.: *Analysis of KNN Information Estimators for Smooth Distributions*. In *IEEE Transactions on Information Theory* 66(6), 2020: pp. 3798–3826

[199] Zinde-Walsh, V.; Galbraith, J. W.: *A Test of Singularity for Distribution Functions*. In *SSRN Electronic Journal* 2011

Eigenständigkeitserklärung
Declaration of Authorship

Hiermit versichere ich an Eides statt, dass die vorliegende Arbeit bisher an keiner anderen Hochschule eingereicht worden ist sowie selbständig und ausschließlich mit den angegebenen Mitteln angefertigt worden ist. Die Stellen der Arbeit, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, sind durch Angaben und Quellen kenntlich gemacht.

Potsdam, 9. Oktober 2023

Johannes Eric Hügle