UNIVERSITY OF AMSTERDAM

MSc ARTIFICIAL INTELLIGENCE
MASTER THESIS

Joint Intervention Detection and Causal Discovery

by
NADJA RUTSCH
13336592

1st February, 2023

48 ECTS
April 2022 - February 2023

Supervisor:
Phillip Lippe

Assessor:
Dr. Sara Magliacane

INFORMATICS INSTITUTE (IvI)
QUVA DEEP VISION LAB (QUVA)

Acknowledgements

First and foremost, I would like to thank my supervisors Sara Magliacane and Phillip Lippe. Thanks for helping me to structure my chaos, enduring my bad taste for "idiodic" abbreviations and, most importantly, for teaching me how to do good research. In particular, I would like to thank Sara for sharing her knowledge in the field of causality with me, equipping me with a deep understanding from which I will benefit for the rest of my academic lifetime. Thanks to Phillip for teaching me how to think like a Neural Network. I owe him credit for contributing a vital part to the loss function of this thesis' main method ALLIN[1].

As the first persons to introduce me to the field of causality, I would like to thank Joris Mooij and Patrick Forré for inspiring me to the topic of this thesis by adding a little philosophical footnote to a collection of otherwise highly formal lecture notes. This footnote[2] caused me to consider the possibility that humans are capable of performing interventions outside of controlled experiments, and that this might affect the data we collect.

Outside of academia, I would like to thank my parents for their emotional and financial support, and for not shaming me for being 30. Thanks to my brother for reminding me that there is life beyond competition, including the academic competition that this thesis is ultimately subject to. And of course, thanks to my friends from the final generation of the UvA's AI Master room, for listening to my frequent rants and providing some more than welcome distraction. Finally, I would like to thank Bart de Rooij for his award-winning massages and his artistic contribution to this thesis. If you find yourself enjoying some of the illustrations, are impressed by the colour of the inner grid in the figure, the subtle off white colouring, the tasteful thickness of the lines... credit is due to him.

[1] this abbreviation turned out ok

[2] "As a less mathematical and more philosophical footnote: it is interesting to speculate about how this [the independence assumption] relates to the notion of a free will. If an agent is not convinced that it chose the intervention values independently of other past aspects of the world, it cannot validly perform this causal reasoning step. [...] So perhaps that is why evolution equipped us with the impression that we have a free will."

Abstract

Observations provide insights about the statistical relations between events, but they can have different causal explanations. Interventions often enable us to differentiate among the different causal structures that are compatible with an observation. However, the presence of interventional data in a dataset might not always be explicitly labeled. In this thesis, we propose a novel differentiable causal discovery method that jointly discovers causal relations and identifies interventions, including their targets. We compare this method to existing causal discovery methods, and to a sequential approach that first separates the data with clustering, and then infers the intervention targets and the causal structure. Experiments on linear Gaussian data with perfect interventions show that both approaches improve over existing methods, which indicates that the methods are generally effective in detecting causal relationships from mixed observational and interventional data with unknown interventions. Our joint approach, in particular, demonstrates a larger potential to exploit hidden interventional information to improve causal discovery.

Table of contents

List of figures

List of tables

Nomenclature

Roman Symbols

$\mathbf{A}$	Adjacency matrix
$\mathbf{C}$	Cluster labels
$\mathcal{C}$	Set of context variables
C	Context variable
D	Number of variables
$\mathcal{E}$	Set of edges
$\mathcal{F}$	Set of structural equations
f	Structural equation
$\mathcal{G}$	Causal graph
$\mathcal{I}$	Set of interventions
I	Intervention
$\boldsymbol{\mathcal{K}}$	Set of clusters
$\mathcal{K}$	Cluster
K	Number of clusters
$\boldsymbol{\mathcal{M}}$	Set of data-generating SCMs
$\mathcal{M}$	Structural causal model
M	Number of datapoints

N	Neighbourhood size
$\mathcal{P}$	Power set
R	Number of intervened variables
$\mathcal{S}$	Conditioning set
$\mathcal{U}$	Set of exogenous variables
U	Exogenous variable
$\mathcal{V}$	Set of vertices (graph) / endogenous variables (SCM)
$\mathbf{W}$	Weight matrix
$\mathbf{X}$	Dataset
X	Endogenous variable
$\mathbf{Z}$	Assignments of datapoints to SCMs

Greek Symbols

α	Significance threshold
λ	Sparsitiy penalty weight
μ	Mean
Ω	Precision matrix
ω	Weight threshold
ρ	Partial correlation coefficient
Σ	Covariance matrix
σ	Standard deviation

Subscripts

$\|\|.\|\|_F$	Fröbenius norm

Other Symbols

$\leftrightarrow$	Bidirected edge

$\rightarrow$ Directed edge

$\odot$ Hadamard product

ℓ Loss function

Acronyms / Abbreviations

ARI Adjusted randomized index

CPDAG Complete Partially Directed Acyclic Graph

CBN Causal Bayesian Network

CI Conditional Independence

DAG Directed Acyclic Graph

EM Expectation Maximization

FN False negatives

FP False positives

GMM Gaussian Mixture Model

MLP Multilayer Perceptron

MEC Markov Equivalence Class

MSE Mean Squared Error

NLL Negative log-likelihood

RI Randomized index

SCM Structural Causal Model

SHD Structural Hamming Distance

1 Introduction

In 1998, a study was published in a medical journal that falsely linked the MMR vaccine to autism (Rao and Andrade, 2011). While the study was quickly found faulty and withdrawn, it still had its effect on public opinion. As a result, vaccination rates declined in some countries like England, where this led to a rising number of measles cases from 56 in 1998 to almost 2,000 in 2012 (UK Health Security Agency, 2022). Many people believe to this day that vaccines cause autism, despite the lack of scientific evidence. One of the drivers for this ongoing misinformation is that the age at which MMR vaccines are administered coincides with the age at which children are typically diagnosed with autism (Chen and DeStefano, 1998).

This is one of many examples where confusion between correlation and causation can have serious negative consequences. Not only humans can fall victim to this confusion, but so can Machine Learning (ML) models. For instance, courts in the United States use the predictive algorithm COMPAS to support judges in their decision making (Johnson et al., 2011). COMPAS predicts how likely a defendant is to become a recidivist, based on factors such as prior crimes, age, or employment status. Even though sensitive attributes like race or gender are not explicitly used for prediction, they can be inferred by the model from other attributes, e.g. ZIP codes which strongly correlate with race (Wiggins, 2020). The per-capita crime rate in the United States is correlated with race, but this does not mean that race is a cause of crime (Sampson et al., 2005). In fact, there is no evidence that an individual's race causes them to be more likely to commit a crime. Yet, COMPAS is more likely to make false positive errors for black defendants and false negative errors for white defendants (Larson et al., 2016), which indicates that the correlation between race and criminal offenses introduces harmful bias into the algorithm's predictions.

When using AI algorithms as tools for decision making, it is therefore crucial to consider the causal perspective. Identifying causal relationships in a system, which is the task of causal discovery, can safeguard decision makers from mistaking correlation for causation. Additionally, causal knowledge makes it possible to predict the effects of potential interventions on a system. For example, predicting an environmental disaster can help to manage the

consequences of the event, but understanding why it occurs makes it possible to prevent it from happening in the first place.

The gold standard of establishing cause-and-effect relationships is through randomized controlled trials. In these trials, treatments are randomly assigned to different groups, such that their effects allow for conclusions about the underlying causal relations. However, randomized controlled trials are often expensive and may not always be feasible to conduct. Alternative methods for determining causality from observational data exist, but either rely on strong assumptions about the underlying data, or are limited in their capabilities.

As motivation for our research, we question an assumption commonly taken for granted: the assumption that the data considered for causal discovery is already split in batches, in which each batch only contains data from the same distribution. In particular, this concerns causal discovery from observational and interventional data. While methods for interventional data with unknown intervention targets exist, they commonnly assume that the data is already separated according to the observational and interventional distributions.

As an example, when collecting data from a system where multiple agents, e.g. humans, can perform interventions, it is possible that samples generated from interventions are present in the data. However, since it is not known which datapoints are interventional, we must identify the interventions to use them for causal discovery. In this thesis, we raise the following question: Can we use interventions to improve causal discovery, even when they are hidden?

One straightforward *sequential* strategy for this is to first identify interventions and then employ existing causal discovery methods. However, we investigate if it is possible to improve upon this, by asking: How can we detect interventions and discover causal relations *jointly*? Based on the assumption that intervention detection can benefit from using an estimate of the causal structure, even if this estimate is imperfect, we propose a joint strategy to intervention detection and causal discovery as a counterpart to the sequential method. To identify which method is most effective in utilizing interventional data to improve causal discovery, we study both approaches in the preliminary context of perfect interventions on linear Gaussian systems that are causally sufficient.

1.1 Contributions

In Chapter 2, we introduce the necessary preliminaries for our work, including causal models, interventions and existing causal discovery algorithms. As part of the existing methods, we cover related work on the topic of causal discovery from mixed datasets with unknown interventions. In Section 2.3, we provide a short overview of the clustering algorithms that we use in the sequential approach. We then formalize the settings of mixed datasets with

unknown interventions in Section 3.1, and present our own methods in Section 3.2 and 3.3 respectively. After this, we evaluate the methods in Chapter 4, where we present our main results in Section 4.2 and additional insights from ablation studies in Section 4.3. Finally, we discuss the implications and future advancements of our work in Chapter 5. Overall, these are our main contributions:

- We present a a sequential method, *PC-JCI with clustering*, that uses the PC algorithm as a constraint-based method for causal discovery from mixed observational and interventional data. In a first phase, our method detects interventions through clustering, and in a second phase, it recovers the causal graph by incorporating cluster memberships into the graph prediction.
- As a method that jointly detects interventions and predicts the causal graph, we propose the differential causal discovery method *ALLIN*. This method performs several iterations of alternating between intervention detection and causal structure learning.
- We show that both methods outperform existing causal discovery methods on linear Gaussian data with perfect interventions. Furthermore, we provide evidence that our joint method *ALLIN* has the larger potential than the sequential method *PC-JCI with clustering* to excel on the setting of mixed observational and interventional datasets with unknown interventions.

2 Background

Our methods, *PC-JCI with clustering* and *ALLIN*, are developed for the task of causal discovery from mixed observational and interventional data. The goal of causal discovery is to identify the cause-and-effect relations in a system. In this chapter, we give an introduction into the field of causality that allows us to define the task of causal discovery formally and measure the effectiveness of our methods. For this, it is necessary to agree on a formal representation of cause-and-effect relations, such as the causal models presented in Section 2.1. This thesis focuses on the setting of mixed observational and interventional datasets with unknown interventions, also called *latent* interventions. In Section 2.1.3, we clarify the notion of an intervention, distinguish it from an observation and explain what it means for an intervention to be latent.

Equipped with this formal language of causality, we offer an overview of existing causal discovery methods in Section 2.2, including one method developed for a similar setting as this thesis. Two of these methods, the PC algorithm (Spirtes et al., 1993) and NOTEARS (Zheng et al., 2018), serve as starting points for the development of our own methods, and are covered in more detail. Since one of our methods, *PC-JCI with clustering*, utilizes clustering, we introduce the relevant clustering algorithms in Section 2.3.

2.1 Causal models

To describe the cause-and-effect relations in a system, several formal frameworks have been developed (Pearl, 2000; Rubin, 1974; Splawa-Neyman et al., 1990). In artificial intelligence, the most widely adopted models are causal Bayesian networks (CBN) and structural causal models (SCM) (Pearl, 2000), the models that we use in this thesis. Both models rely on complementary graph representations to describe the existence of causal effects between variables.

2.1.1 Graph representation

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by a set of D vertices, also called nodes, $\mathcal{V} = \{X_1, ..., X_D\}$ and a set of edges $\mathcal{E}$ that connect the vertices. In a causal graph, we assume that each vertex X_i corresponds to a random variable X_i. In this case, a directed edge from vertex X_i to X_j indicates that X_i is a direct cause of X_j. Regarding terminology, describing vertex-relations in graphs is similar to describing a family tree: a variable X_i with an incoming edge from X_j is a *child* of X_j in $\mathcal{G}$, and correspondingly X_j is the *parent* of X_i. Each parent of X_i is at the same time an *ancestor* of X_i. Recursively, all ancestors of parents of X_i are also ancestors of X_i. If X_j is an ancestor of X_i, it follows that X_i is a *descendant* of X_j.

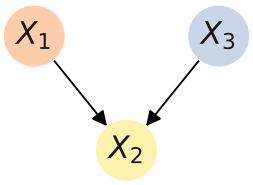


Fig. 2.1 Example of a collider structure. X_2 is a collision node on the path between X_1 and X_3.

Generally, all vertices that are connected by an edge are *adjacent*. Adjacencies create *paths* between vertices in the graph. A path between X_i and X_j is a sequence of distinct vertices $(X_i, ..., X_j)$ such that each two consecutive vertices are adjacent. To create a graph $\mathcal{G}_{skel}$ that describes the adjacencies of a graph $\mathcal{G}$, all directed edges are transformed to undirected edges. The resulting graph is the *skeleton* of the directed graph.

To describe the edge orientations on a 3-variable path $X_i - X_j - X_k$, we distinguish between a *chain*, *fork* and a *collider*. The graph structure $X_i \to X_j \to X_k$ and its permuted version $X_k \to X_j \to X_i$ are both chains. When X_j causes X_i and X_k, the corresponding graph structure $X_i \leftarrow X_j \to X_k$ is a fork. The collider structure $X_i \to X_j \leftarrow X_k$ includes a collision node X_j that receives incoming edges from two variables (see Figure 2.1). The collider is *unshielded* if X_i and X_k are not adjacent in the context of the full causal graph.

Different classes of graphs can be used to represent a causal model. The simplest of these classes is the Directed Acyclic Graph (DAG), which does not contain any cycles and only a certain type of edge, the directed edge ($\to$). In causal discovery algorithms, a common assumption is *causal sufficiency*, i.e. the true underlying graph is assumed to be a DAG. For the scope of this thesis, we also assume the system is causally sufficient, i.e. we restrict ourselves to the prediction of DAGs.

For models that can be described by a DAG, the joint probability distribution of the variables can be factorized according to the graph. Each variable X_i is conditioned on its parent variables Pa_i:

$$P(X_1, X_2, ..., X_D) = \prod_{i=1}^{D} P(X_i | \mathrm{Pa}_i) \tag{2.1}$$

This factorization is possible due to the conditional independencies that are implied by the DAG. To derive conditional independencies, a DAG can be characterized by the d-separations it contains. In correspondence with their implied independence relations, d-separations can be conditioned on any subset $\mathcal{S} \in \mathcal{P}(\mathcal{V})$. Marginal independencies are expressed by choosing $\mathcal{S} = \varnothing$ and are implied by the unconditional d-separations of the graph.

Definition 2.1.1 (d-separation)**.** *Two vertices X_i and X_j are d-separated by $\mathcal{S}$ if and only if all existing paths between X_i and X_j are blocked by $\mathcal{S}$.*

Trivially, all vertex-pairs without existing paths between them are d-separated in $\mathcal{G}$, independent of the conditioning set $\mathcal{S}$. Beyond the graph's adjacencies, the presence of *unshielded colliders* determines which paths are blocked, and hence, which vertices are d-separated. In the unconditional case $\mathcal{S} = \varnothing$, a path is blocked if and only if it contains a collider. For the general case, we follow Pearl et al. (2016) by defining:

Definition 2.1.2 (Blocked path)**.** *A path is blocked by $\mathcal{S}$ if and only if:*

1. *the path contains a chain of nodes $X_i \to X_j \to X_k$ or a fork $X_i \leftarrow X_j \to X_k$ such that the middle node X_j is in $\mathcal{S}$, or*
2. *the path contains a collider $X_i \to X_j \leftarrow X_k$ such that the collision node X_j is not in $\mathcal{S}$, and no descendant of X_j is in $\mathcal{S}$.*

An existing d-separation by $\mathcal{C}$ between vertex X_i and X_j implies the following conditional independence:

$$P(X_i, X_j \mid \mathcal{S}) = P(X_i \mid \mathcal{S}) P(X_j \mid \mathcal{S}) \tag{2.2}$$

Faithfulness. Under the Causal Faithfulness Assumption (Spirtes et al., 1993), conditional independencies in the joint probability distribution imply d-separations in the underlying causal graph. This assumption allows to infer properties of the causal graph from conditional independencies. For example, the conditional independence in Equation 2.2 implies that there exists a d-separation between X_i and X_j conditioned on $\mathcal{S}$, which means that X_i and X_j can not be adjacent. This implication is used in a number of causal discovery methods, such as the PC algorithm.

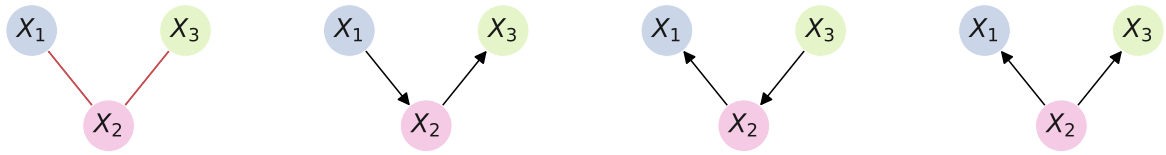
Causal sufficiency. The middle node X_j of a fork $X_i \leftarrow X_j \to X_k$ can also be called a *confounder* of X_i and X_j, as it is a common cause of both variables. If X_j is not measured as a part of the system, it is a *latent confounder*. To represent various forms of confounding between a variable-pair (X_i, X_k), the variables can be connected with a bidirected edge ($\leftrightarrow$). A graph is

causally sufficient and does not have any bidirected edges if all common causes of pairs in $\mathcal{V}$ are also in $\mathcal{V}$, and there is no selection bias. Many causal discovery methods assume that the true underlying graph is causally sufficient. For simplicity, this thesis also focuses on causally sufficient graphs.

Markov Equivalence Class

Even when the true underlying causal graph is a DAG, some causal discovery methods do not always return a DAG, e.g. if they can not determine the direction of a causal effect between two variables. To represent uncertainty about the direction of causal relations, we use mixed graphs, where undirected edges connect the affected vertex-pairs. A subset of mixed graphs belongs to the class of Complete Partially Directed Acyclic Graphs (CPDAG). Mixed graphs are CPDAGs if they represent the Markov Equivalence Class (MEC) of a causal DAG.

MECs contain DAGs that have the same d-separations, i.e. DAGs with identical adjacencies and identical unshielded colliders. Figure 2.2 shows the CPDAG of an MEC with three vertices $\mathcal{V} = \{X_1, X_2, X_3\}$ and two undirected edges $\mathcal{E} = \{(X_1, X_2), (X_2, X_3)\}$, and the DAG members of this MEC. Since these DAGs have the same adjacencies and no unshielded collider, they belong to the same MEC. Note that the graph $X_1 \to X_2 \leftarrow X_3$ is not a member of the DAG. As opposed to the graphs that belong to the MEC, the vertices X_1 and X_3 are not d-separated by X_2 in this graph, as conditioning on the collision node X_2 opens up the path between X_1 and X_3.



(a) CPDAG. (b) $\mathcal{G}_1$. (c) $\mathcal{G}_2$. (d) $\mathcal{G}_3$.

Fig. 2.2 Graphs that belong to the same MEC and the CPDAG that represents the MEC. Undirected edges are coloured red.

Ultimately, a MEC provides information about the identifiability of the causal graph from observational data, i.e. data that is sampled from the joint probability distribution. While the skeleton is, in theory, always identifiable under the faithfulness assumption, only the edge directions that are shared among all of the graphs in the same MEC can be identified from observational data. Under the assumption of faithfulness, all d-separations of the graph can be identified in the limit of infinite observational data, and the true graph must be a member of the MEC defined by these separations.

2.1.2 Structural causal model

In this thesis, we rely on structural causal models (SCMs) to describe a causal system. In the SCM framework, a causal system is defined by a set of structural equations $\mathcal{F}$. Additionally, an SCM $\mathcal{M} = (\mathcal{V}, \mathcal{U}, \mathcal{F})$ distinguishes between endogenous and exogenous causal variables, defined in the sets $\mathcal{V}$ and $\mathcal{U}$ respectively. Each endogenous causal variable is annotated with its own structural equation that describes how its value can be calculated from its causes. To account for randomness or noise, variables in the exogenous set $\mathcal{U}$ are left unexplained by the causal model. If the SCM is Markovian, the exogenous variables are jointly independent, such that each exogenous variable appears in at most one structural equation. For a Markovian SCM $\mathcal{M}$, exogenous variables are usually omitted in the graph representation $\mathcal{G}(\mathcal{M})$ of the model, without affecting any d-separations between endogenous variables.

As an example, consider the following Markovian SCM with two endogenous variables and two exogeneous terms:

$$\mathcal{V} = \{X_1, X_2\} \qquad \mathcal{U} = \{U_1, U_2\} \qquad \mathcal{F} = \{f_1, f_2\} \tag{2.3}$$

$$f_1 : X_1 = U_1$$
$$f_2 : X_2 = X_1 + U_2$$

Figure 2.3 shows the graph that corresponds to this SCM. Since X_1 causes X_2, the graph contains a directed edge from X_1 to X_2. The exogenous terms U_1 and U_2 are omitted in $\mathcal{G}(\mathcal{M})$, as each term only appears in one structural equation. For Gaussian distributed exogenous random variables U_1 and U_2, the example model describes a *linear Gaussian* system.

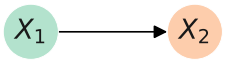


Fig. 2.3 Graph representation $\mathcal{G}(\mathcal{M})$ of the SCM; exogenous variables ommitted.

2.1.3 Interventions

In the language of causality, a randomized controlled trial involves an *intervention*. For instance, in a medical trial evaluating the effect of an antidepressant, an intervention is performed on antidepressant intake. An intervention $I \in \mathcal{P}(\mathcal{V})$ is an event where each variable $X_i \in I$, such as the antidepressant intake, is pertubed. Formally, this means that the underlying data-generating process $\mathcal{F}^{(I)}$ differs from the process $\mathcal{F}$ that generates data from the joint observational distribution. In our example, antidepressant intake is usually influenced by various factors, such as a patient's demographics or the severity of their depression. In the randomized controlled

trial, these influences are removed, and the treatment is randomly assigned, independent of the factors that usually influence it.

To describe a causal system under an intervention I formally, we can make use of the SCM framework, and define an interventional SCM $\mathcal{M}^{(I)} = (\mathcal{V}, \mathcal{U}^{(I)}, \mathcal{F}^{(I)})$ as a modification of the observational SCM $\mathcal{M}$. Generally, we define the structural equation of each endogenous variable $f_i^{(I)} \in \mathcal{F}^{(I)}$ anew. However, not every structural equation necessarily differs from its definition in the observational SCM $\mathcal{M}$. Only the variables, for which the structural equations are different under intervention I, are considered a *target* of intervention I. An intervention can either occur on a single target, i.e. $|I| = 1$, or on multiple targets simultaneously. As an example of an intervention on the SCM described in Equations 2.3, we define the following SCM $\mathcal{M}^{(I)} = (\mathcal{V}, \mathcal{U}^{(I)}, \mathcal{F}^{(I)})$, where the intervention I targets variable X_2, such that $f_2^{(I)} \neq f_2$:

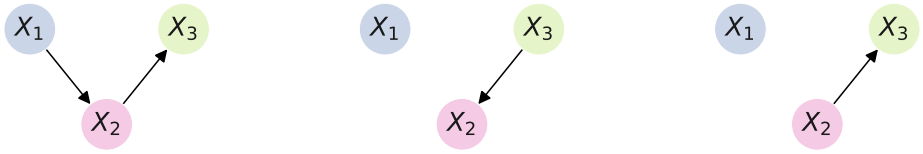
$$I = \{X_2\} \qquad \mathcal{V} = \{X_1, X_2\} \qquad \mathcal{U}^{(I)} = \{U_1, U_2^{(I)}\} \qquad \mathcal{F}^{(I)} = \{f_1^{(I)}, f_2^{(I)}\} \tag{2.4}$$

$$f_1^{(I)} : X_1 = U_1$$
$$f_2^{(I)} : X_2 = U_2^{(I)}$$

Commonly, an intervention is assumed to modify existing dependencies between targeted variables and their causes, and not add new dependencies. For the associated graph representation $\mathcal{G}^{(I)}$ of the endogenous variables, this means that an intervention can only remove edges, either all incoming edges to the target variable (*perfect* intervention) or a subset of all incoming edges (*imperfect* intervention).

$\mathcal{I}$-Markov equivalence

Since an intervention only affects incoming edges but leaves outgoing edges intact, the behaviour of the system under an intervention can help to identify the direction of a causal effect. Hence, data from interventional settings has the potential to increase the identifiability of edge directions in the causal structure beyond the MEC. Under the assumption of faithfulness, dependencies that remain under an intervention imply remaining d-separations in the graph. This information enables the exclusion of more members from the MEC as possible causal DAG candidates. From knowing the MEC and the d-separations of the interventional DAGs for perfect interventions $\mathcal{I} = \{I_1, I_2, ..., I_R\}$, possible candidates can be reduced to members of the $\mathcal{I}$-Markov equivalence class (Hauser and Bühlmann, 2012). Two DAGs $\mathcal{G}_i$ and $\mathcal{G}_j$ are $\mathcal{I}$-Markov equivalent, if they belong to the same MEC and if the interventional graphs $\mathcal{G}_i^{(I)}$ and $\mathcal{G}_j^{(I)}$ have the same adjacencies for all $I \in \mathcal{I}$.



(a) $\mathcal{G}_1^{(I)}$. (b) $\mathcal{G}_2^{(I)}$. (c) $\mathcal{G}_3^{(I)}$.

Fig. 2.4 Graphs from the same MEC under the perfect intervention $I = \{X_1\}$. All incoming edges to X_1 are removed.

For the MEC in Figure 2.2, the intervention $I = \{X_1\}$ results in the interventional graphs shown in Figure 2.4. Since $\mathcal{G}_1^{(I)}$ has unique adjacencies within its MEC, it can be identified from this intervention.

Latent interventions

If interventions are present in the data, the dataset is generated from a mixture of SCMs $\boldsymbol{\mathcal{M}} = \{\mathcal{M}^{(\varnothing)}, \mathcal{M}^{(I_1)}, ..., \mathcal{M}^{(I_R)}\}$ with R different interventions. To emphasize that the observational SCM is not subject to any interventions, we denote it as $\mathcal{M}^{(\varnothing)}$. Each SCM $\mathcal{M}^{(I_r)}$ denotes the interventional SCM under intervention I_r. In a dataset of M datapoints with D variables $\mathbf{X} \in \mathbb{R}^{M \times D}$, each datapoint $\mathbf{x}_m$ is generated by one SCM. To describe which datapoint is generated by which SCM, we denote the one-hot encoded assignments for the whole dataset as $\mathbf{Z} \in \mathbb{R}^{M \times (R+1)}$.

Commonly, causal discovery methods assume that the data is separated into batches according to the observational and interventional distributions and that it is known which batches are interventional (Brouillard et al., 2020; Hauser and Bühlmann, 2012; Lippe et al., 2022; Mooij et al., 2020). The targets of interventional datapoints can be either known or unknown. However, it is also possible that interventional data is present in a given data collection, but not labelled as such. Interventions where both $\boldsymbol{\mathcal{M}}$ and $\mathbf{Z}$ are unknown are considered to be *fully latent* (Faria et al., 2022). Our own proposed causal discovery methods are developed for data collections that may include latent interventions. In the following section, we will describe a selection of existing causal discovery methods in detail, including methods for observational data, data with known interventions and targets, as well as one method for latent interventions.

2.2 Causal structure learning

The task of causal structure learning, or causal discovery, is to recover the graph representation of a causal system, up to the MEC or $\mathcal{I}$-MEC if interventional data is available. Since the

search space of potential graph candidates is discrete and grows super-exponentially with the number of nodes, finding the most suitable causal graph is a difficult combinatorial optimization problem.

Traditionally, causal discovery methods are either constraint-based, score-based or hybrid (Glymour et al., 2019). Constraint-based methods like PC (Spirtes et al., 1993) and Fast Causal Inference (FCI) (Spirtes et al., 2000) perform causal structure search by ensuring the satisfaction of constraints that are derived from conditional independence testing. Score-based algorithms, such as Greedy Equivalence Search (GES) (Chickering, 2002), perform causal discovery by optimizing a score function like BIC. In line with recent advancements in gradient-based optimization, numerous differentiable causal discovery methods have been proposed (Brouillard et al., 2020; Lippe et al., 2022; Zheng et al., 2018). As a framework that can be applied to various causal discovery methods, Joint Causal Inference (JCI) (Mooij et al., 2020) includes additional context variables into the causal search, which can be used to infer intervention targets. We call the application of JCI to the PC algorithm *PC-JCI*. Faria et al. (2022) have developed a causal discovery method that focuses on a similar setting as this thesis. Since this method modifies the method *Differential Causal Discovery from Interventional Data* (DCDI) (Brouillard et al., 2020) for latent interventions, we call it DCDI-L. Table 2.1 shows an overview of different assumptions of the relevant causal discovery methods PC, PC-JCI, NOTEARS, ENCO and DCDI-L.

Table 2.1 Assumptions of the constraint-based methods PC and PC-JCI, and the score-based differentiable methods NOTEARS, ENCO and DCDI-L.

Method	**Faithfulness**	**Data distributions**	**Interventional data**
PC (Spirtes et al., 1993)	Yes	Any	None
PC-JCI (Mooij et al., 2020)	Yes	Any	Unknown targets
NOTEARS (Zheng et al., 2018)	No	Linear Gaussian	None
ENCO (Lippe et al., 2022)	No	Any	Known targets
DCDI-L (Faria et al., 2022)	No	Any	Latent

2.2.1 PC

The PC algorithm infers the causal structure from constraints given by the d-separations of the graph. Based on the assumption of faithfulness, d-separations are derived from conditional independence tests. Two conditional distributions are independent if they satisfy Equation 2.2. In practice, the true probability distributions are not known and can only be estimated empirically through samples from the distributions. Hence, distributions can not be checked for conditional independencies via Equation 2.2, but require the usage of empirical tests.

Conditional independence (CI) testing is notoriously difficult: it is significantly harder to test than unconditional independence, especially for continuous conditioning variables (Bergsma, 2004; Shah and Peters, 2020).

Fisher's z-transformation

A variety of CI tests have been developed to account for different data types, dependencies and distributions (Bellot and van der Schaar, 2019; Fisher, 1915; Shah and Peters, 2020; Strobl et al., 2019; Zhang et al., 2012). If the underlying data distribution is multivariate Gaussian, Fisher's z-transformation gives a good approximation of the probability of the data under the null hypothesis of conditional independence (Fisher, 1915). The transformation is based on the sample partial correlation coefficient $\widehat{\rho}_{X_i X_j|\mathcal{S}}$ between variable X_i and X_i conditioned on $\mathcal{S}$, where $\widehat{p}_{ij}$ is element ij of the precision matrix $\Omega = \Sigma^{-1}$ of the joint distribution of X_i, X_j and all $X \in \mathcal{S}$:

$$\widehat{\rho}_{X_i X_j|\mathcal{S}} = -\frac{\widehat{p}_{ij}}{\sqrt{\widehat{p}_{ii}\widehat{p}_{jj}}} \tag{2.5}$$

Applying the Fisher z-transformation on this sample partial correlation coefficient $\widehat{\rho}_{X_i X_j|\mathcal{S}}$ yields a value z that is approximately normally distributed for any bivariate Gaussian variables X_i and X_j:

$$z(\widehat{\rho}_{X_i X_j|\mathcal{S}}) = \frac{1}{2}\ln\left(\frac{1+\widehat{\rho}_{X_i X_j|\mathcal{S}}}{1-\widehat{\rho}_{X_i X_j|\mathcal{S}}}\right) \tag{2.6}$$

The distribution of z depends on the true correlation ρ and the sample size M, approximately giving the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with the following mean and variance for independent variables ($\rho = 0$):

$$\mu = \frac{1}{2}\ln\left(\frac{1+\rho}{1-\rho}\right) = 0 \qquad\qquad \sigma^2 = \frac{1}{M-3} \tag{2.7}$$

By assuming that the true correlation is $\rho = 0$, the probability of the sample partial correlation coefficient $\widehat{\rho}_{X_i X_j|\mathcal{S}}$ under the null hypothesis can be computed. The null hypothesis is rejected if this probability is lower than a certain threshold α, typically 0.01 or 0.05, and X_i and X_j are then assumed to be dependent given $\mathcal{S}$.

Algorithm

The original version of the PC algorithm infers the causal structure in the following three steps (Spirtes et al., 1993):

1. The skeleton of the graph is predicted.

For this, the algorithm starts from the assumption of a *complete* graph, i.e. a graph in which every vertex-pair is connected by an undirected edge. An edge between X_i and X_j is removed if there exists a conditioning set $\mathcal{S}$ such that Equation 2.2 holds, according to a CI test like Fisher's z-test. Since $\mathcal{S} \in \mathcal{P}(\mathcal{V} \backslash \{X_i, X_j\})$, there are 2^{D-2} possible conditioning sets for each variable-pair. Hence, the algorithm tries to reduce the number of CI tests as much as possible.

At $t = 0$, it starts with the smallest conditioning set $\mathcal{S} = \varnothing$ and tests all variable-pairs for marginal independencies, removing an edge after each test that fails to reject the null hypothesis. After performing all necessary tests for conditioning sets of size $|\mathcal{S}| = t$, the algorithm moves on to larger conditioning sets of size $|\mathcal{S}| = t + 1$. Which tests are necessary is determined as follows: (1) only vertex-pairs that are still adjacent at the time of the test need to be considered and (2) for any given vertex-pair X_i and X_j, it is only necessary to test for conditioning sets $\mathcal{S}$ for which all vertices in $\mathcal{S}$ are adjacent to both X_i and X_j.

The algorithm stops when no more tests are necessary, which happens when the largest neighbourhood in the remaining graph $N_{\max}$ is smaller than the minimum size of the leftover conditioning sets. As a preparation for the next step, the algorithm keeps track of all encountered separating sets for each vertex-pair.

2. Unshielded colliders are identified and their edges oriented.

Unshielded colliders are uniquely identifiable by their d-separations: their parents are d-separated by some set $\mathcal{S} \in \mathcal{P}(\mathcal{V})$, but they are not d-separated when conditioned on the collider. Therefore, each unshielded triple in $\mathcal{G}$, i.e. each triple $X_i - X_j - X_k$ where X_i and X_k are not adjacent, is checked for the corresponding pattern of conditional independence. If the middle variable X_j was not recorded as a separating set of X_i and X_k in step 1, the triple must be a collider and can be oriented.

3. As many of the remaining edges as possible are oriented with Meek's rules (Meek, 1995).

The application of Meek's rules (Meek, 1995) ensures that the predicted graph is a DAG and that no additional unshielded colliders are introduced. Since only triples with the mentioned conditional indepedence pattern can be colliders, all edges where one direction would result in a new unshielded collider but the pattern of independencies is missing, can be oriented into the

other direction. Additionally, edges where one direction would result in a cycle can be oriented in the other direction, as the PC algorithm assumes that the true causal graph is a DAG. Meek's rules are shown to be sound and complete, which means they do not introduce any error (sound) and find all possible orientations in the MEC (complete).

Given the correct conditional independencies, the PC algorithm returns the MEC, independent of the order in which the vertex-pairs and conditioning sets in step 1 are tested. In practice, this order becomes relevant. Since results of CI tests are prone to error, mistakenly removing an edge will affect the following tests – the more, the earlier the edge is removed. To yield an order-independent skeleton, a stable version of the PC algorithm has been developed, *PC-stable* (Colombo et al., 2014). In this version, edges between two vertices X_i and X_j are only removed after testing for all conditioning sets of the same size. Throughout the thesis, we use this version of PC as our baseline and the basis for our sequential method.

2.2.2 JCI

The framework of Joint Causal Inference (JCI) (Mooij et al., 2020) allows for joint inference of intervention targets and the causal graph. For this, the interventional identity of datapoints can be expressed in the form of *context variables*. By augmenting datapoints with a one-hot encoding of their context memberships, these context variables can be included in the causal structure search, even without knowing the intervention target of a context variable. To improve the identifiability of the extended causal graph, background knowledge of varying degrees can be incorporated. The JCI framework distinguishes between the following three *JCI assumptions* on background knowledge:

1. Context variables are not caused by non-context variables.
2. Context variables are not confounded with non-context variables.
3. All context variables are confounded but are not direct causes of each other.

The framework is not restricted to a particular algorithm, but can extend existing constraint-based causal search methods like PC and FCI. By running the causal search method of choice on the extended dataset, edges between context variables and original variables can be inferred. In the context of interventions, an edge from context variable C_i to variable X_j would indicate that X_j is an intervention target of all datapoints for which $C_i = 1$, or of all datapoints for which $C_i = 0$.

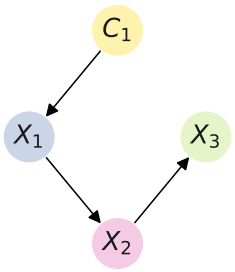


Fig. 2.5 $\mathcal{G}_1$ with context variable C_1.

For an extension of the example graph $\mathcal{G}_1$ (Figure 2.2b, 2.4a) with a context variable, see Figure 2.5. In this example, the data contains

context information about variable X_1, e.g. information about interventions on X_1. For the graph in Figure 2.5, JCI assumptions (1) and (2) hold, and assumption (3) holds trivially, as there is only one context variable. As part of our own method *PC-JCI with clustering*, we use the JCI framework in combination with the PC algorithm, where we discover an extended causal graph with PC.

2.2.3 NOTEARS

Non-combinatorial Optimization via Trace Exponential and Augmented lagRangian for Structure learning (NOTEARS) (Zheng et al., 2018) is a differential causal discovery method that formulates the causal search problem as a continuous optimization problem. Instead of directly searching the adjacency matrix $\mathbf{A} \in \{0,1\}^{D\times D}$ of the graph, it aims to find the weight matrix $\mathbf{W} \in \mathbb{R}^{D\times D}$ of the structural equations, i.e. the weights of causes on their effects. For linear dependencies, the predicted variables are simply $\widehat{\mathbf{X}} = \mathbf{XW}$, such that the fit of $\mathbf{W}$ can be judged using the following mean squared error (MSE):

$$\ell(\mathbf{W};\mathbf{X}) = \frac{1}{2M}\|\mathbf{X} - \mathbf{XW}\|_F^2 \tag{2.8}$$

To prevent the prediction of variables from themselves, the diagonal of $\mathbf{W}$ is set to 0. For the remaining weights, a weight regularization term is included in the objective $F(\mathbf{W})$ to encourage sparseness:

$$F(\mathbf{W}) = \ell(\mathbf{W};\mathbf{X}) + \lambda\|\mathbf{W}\|_1 \tag{2.9}$$

Optimizing $F(\mathbf{W})$ alone does not yet achieve the desired result, because the result would include weights for predicting causes from effects. This, in turn, would yield a weight matrix that corresponds to a cyclic graph. To impose a causal order on $\mathbf{W}$, Zheng et al. (2018) incorporate the following acyclicity constraint in the objective:

$$h(\mathbf{W}) = \operatorname{tr}\left(e^{\mathbf{W}\odot\mathbf{W}}\right) - D = 0 \tag{2.10}$$

The value of $h(\mathbf{W})$ serves as a measure of cyclicity, and is only 0 if $\mathbf{W}$ is acyclic. While the acyclicity constraint imposes a causal order on the result, it does not guarantee that this order is

the correct one. With this constraint, Zheng et al. (2018) arrive at the following final objective:

$$\begin{aligned} \widehat{\mathbf{W}} = &\min_{\mathbf{W}\in\mathbb{R}^{D\times D}} F(\mathbf{W}) \\ &\text{subject to } h(\mathbf{W}) = 0 \end{aligned} \tag{2.11}$$

Deriving the adjacency matrix $\widehat{\mathbf{A}}$ from the weight matrix $\widehat{\mathbf{W}}$ is done by choosing a threshold ω for the coefficients, such that:

$$\widehat{A}_{ij} = \begin{cases} 1 & \text{if } |\widehat{W}_{ij}| > \omega \\ 0 & \text{otherwise} \end{cases} \tag{2.12}$$

Optimization

Since the objective is subject to a constraint, it can not directly be solved with gradient descent methods. The application of gradient-based optimization is facilitated by reformulating the problem as a dual function with Lagrangian multipliers ρ and α:

$$D(\alpha) = \min_{\mathbf{W}\in\mathbb{R}^{D\times D}} F(\mathbf{W}) + \frac{\rho}{2}|h(\mathbf{W})|^2 + \alpha h(\mathbf{W}) \tag{2.13}$$

To avoid that the objective is dominated by the acyclicity terms, the Lagrangian multipliers are chosen as small as possible to satisfy the constraint up to some tolerated value of $h(\mathbf{W})$. Accordingly, the algorithm starts with a small initial value of α and alternates between updating the weight matrix $\mathbf{W}$ and increasing the Lagrangian multiplier α. The multiplier ρ is chosen such that a minimal improvement in acyclicity is achieved after the update of $\mathbf{W}$. It serves as a step size of updating α, ensuring linear convergence of α to the value that would maximize $D(\alpha)$:

$$\alpha_{t+1} = \alpha_t + \rho\nabla_\alpha D(\alpha) = \alpha_t + \rho h(\mathbf{W}_t) \tag{2.14}$$

2.2.4 ENCO

The differentiable causal discovery method *Efficient Neural Causal discOvery* (ENCO) (Lippe et al., 2022) learns the causal graph by alternating between learning the structural equations from observational data and fitting the graph to interventional data. The inputs to the structural equations are masked according to the learned graph at the time of the iteration, and the structural functions are fixed when updating the graph. For learning the adjacency matrix, ENCO models the existence of edges separately from their direction, where the probabilities of

both directions must sum up to one, i.e. $P(X_i \to X_j) = 1 - P(X_j \to X_i)$. Under the assumption that edge directions can only reliably be predicted from interventions, the direction probabilities of an edge are only updated after encountering interventional data on the incoming or outgoing vertex of the edge. ENCO is developed for datasets with known interventions on each variable and known intervention targets, but can also be applied to datasets with interventions on fewer variables.

2.2.5 DCDI-L

Faria et al. (2022) have developed a variational approach to the problem of causal discovery from data with latent interventions, which we will call DCDI-L in this thesis. DCDI-L extends the differentiable causal discovery method DCDI (Brouillard et al., 2020) to also optimize the assignments $\mathbf{Z}$ of datapoints as coming from the observational or interventional distributions. A potentially unspecified number of different interventional distributions is modelled as being generated by a Dirichlet process prior. In line with common assumptions on interventions, the parents of variables in an intervened SCM are restricted to subsets of the parents in the observational SCM. Under this assumption, DCDI-L does not require interventions to be perfect, but rather allows to model perfect and imperfect interventions. In this thesis, we will only consider perfect interventions.

2.3 Clustering

A clustering algorithm is used in the first phase of our method *PC-JCI with clustering*. The goal of clustering is to find natural groupings in data, such that items within each cluster are more similar to each other than they are to items in other clusters. Its ability to identify underlying patterns in data makes clustering an interesting technique for separating observational and interventional datapoints according to their generating SCMs. One of the best-known clustering algorithms is k-means (Lloyd, 1982) which is popular due to its simplicity and efficiency. If the true clusters are multivariate Gaussians, a Gaussian mixture model (GMM) is suitabe to learn the clustering (Mclachlan and Basford, 1988). This is the case for the linear Gaussian data that we study in this thesis, such that we use the GMM as the main clustering algorithm for our method *PC-JCI with clustering*.

2.3.1 k-means

k-means clustering divides a dataset into a specified number k of clusters. This is achieved by iteratively assigning each datapoint to the cluster with the nearest mean, and then updating the

cluster means to the average of the items in the cluster. With these steps, the algorithm is a special case of the expectation maximization (EM) algorithm. For the first iteration, cluster means are either initialized randomly or in a way that maximizes the distance between the centers (Arthur and Vassilvitskii, 2007).

2.3.2 Gaussian mixture model

A GMM is a probabilistic model that assumes that the underlying data is generated by a mixture of Gaussian distributions. This means that each datapoint is assumed to come from one of a fixed number of underlying Gaussian distributions, with the probability of a datapoint belonging to each distribution determined by a set of mixture coefficients. By fitting a GMM to a dataset, the mixture coefficients and the parameters of the underlying Gaussian distributions can be estimated, i.e. their means and covariances. After fitting the GMM, labels for each datapoint can be derived by choosing the cluster with the probability given the value of the datapoint and the mixture coefficients. It is also possible to obtain soft cluster labels by directly using the probabilities over labels for each data point, instead of rounding the probabilities to hard assignments. An advantage of GMMs is that they can handle a larger variety of cluster shapes as compared to k-means. Figure 2.6 shows an example where k-means fails and the GMM performs well.

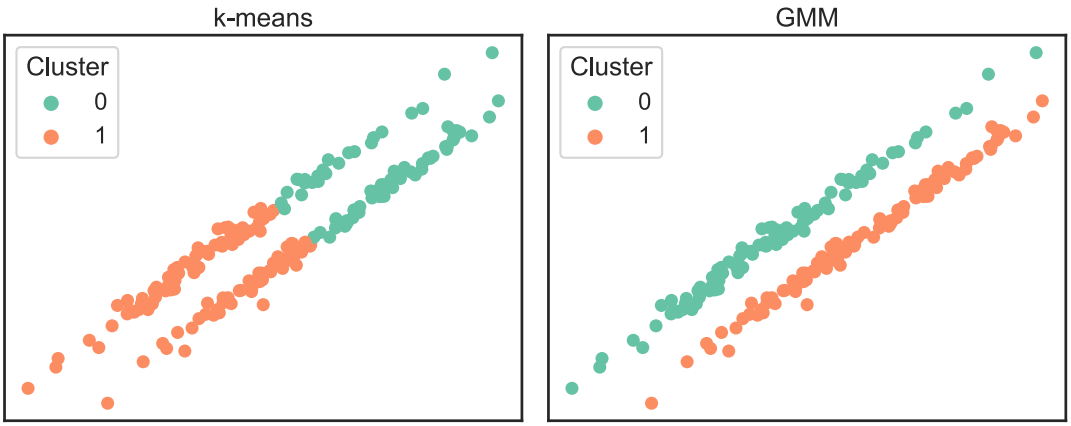


Fig. 2.6 Comparison of k-means (left) and GMM clustering (right). The GMM is able to model clusters of the depicted shape, while k-means is not.

3 Joint Intervention Detection and Causal Discovery

In this chapter, we present two methods for causal discovery on mixed observational and interventional datasets with unknown interventions, *PC-JCI with clustering* and *ALLIN*. First, we define the specific setting for which our methods are developed in Section 3.1. Then, we introduce *PC-JCI with clustering* in Section 3.2. This method follows a sequential approach to intervention detection and causal discovery, where interventions are detected through clustering, and causal discovery is performed with the constraint-based PC algorithm. In Section 3.3, we present *ALLIN*, a fully differentiable method that jointly detects interventions and predicts the causal structure.

3.1 Setting

To describe the causal system in its observational state, we consider a Markovian SCM $\mathcal{M}^{(\varnothing)}$ with D continuous variables $\mathcal{V} = \{X_1, ..., X_D\}$. On this SCM, R different interventions $I_1, ..., I_R$ have been performed, which we describe with the interventional SCMs $\mathcal{M}^{(I_1)}, ..., \mathcal{M}^{(I_R)}$. Generated from this family of SCMs $\boldsymbol{\mathcal{M}} = \{\mathcal{M}^{(\varnothing)}, \mathcal{M}^{(I_1)}, ..., \mathcal{M}^{(I_R)}\}$, we have a data collection $\mathbf{X} \in \mathbb{R}^{M \times D}$ of M datapoints.

We assume that the observational SCM is acyclic, which means that the graph representation of the SCM is also acyclic. Additionally, we assume that the system is causally sufficient, i.e. that all common causes of variables $X \in \mathcal{V}$ are also in $\mathcal{V}$ and there is no selection bias. It follows that the underlying causal graph is a DAG. We consider interventions $I_1, ..., I_R$ to be perfect, i.e. they remove any causal parents of the intervened variables and impose an independent distribution on each intervened variable. Hence, acyclicity and causal sufficiency of the interventional SCMs are implied by the same assumptions for the observational case.

For the scope of this thesis, we restrict ourselves to linear Gaussian systems, i.e. structural equations that are linear functions of the variables and exogenous terms that are Gaussian

distributed. The interventions can occur on a single target or on multiple targets simultaneously. Most importantly, we assume that the interventions in our data collection are latent, and hence we neither know the datapoints' assignments $\mathbf{Z}$ to the SCMs in $\boldsymbol{\mathcal{M}}$, nor $\boldsymbol{\mathcal{M}}$ itself. This also means that the size of $\boldsymbol{\mathcal{M}}$ is not known. While, in our experiments, we focus on the setting where single-target interventions on each variable are present in the data, i.e. $R = D$, our methods are not restricted to this setting.

3.2 PC-JCI with clustering

PC-Joint Causal Inference (PC-JCI) with clustering performs intervention detection and causal discovery sequentially. Figure 3.1 illustrates the process of combining PC-JCI with clustering. At first, we apply a clustering algorithm to the data (Figure 3.1, left side), resulting in Kclusters $\boldsymbol{\mathcal{K}} = \{\mathcal{K}_1, ..., \mathcal{K}_K\}$. The goal of the clustering is to separate the datapoints according to their generating SCMs $\mathcal{M} \in \boldsymbol{\mathcal{M}}$. Ideally, datapoints which were generated under the same interventions should be assigned to the same cluster. However, even if the clustering algorithm identifies the ground truth clusters, the intervention targets of each cluster are still unknown.

To find the intervention targets, we apply the JCI framework (Mooij et al., 2020) to the PC algorithm. We encode the cluster memberships as context variables $\mathcal{C} = \{C_1, ..., C_K\}$ (Figure 3.1, right side) in the JCI framework. This is done by concatenating the one-hot encoded cluster labels $\mathbf{C} \in \{0,1\}^{M \times K}$ to the normalized features $\mathbf{X} \in \mathbb{R}^{M \times D}$, resulting in the following augmented data $\mathbf{X}' \in \mathbb{R}^{M \times (D+K)}$:

$$\mathbf{X}' = \mathbf{X} | \mathbf{C} \tag{3.1}$$

We proceed by giving the augmented data $\mathbf{X}'$ as input to the PC algorithm. By testing for conditional independencies, the PC algorithm returns a prediction of the extended causal graph, including the context variables. To arrive at the final prediction of the causal graph, the context variables and all edges adjacent to them are removed from the prediction.

It is important to note that augmenting the data with such cluster labels introduces faithfulness violations into the system. Due to the one-hot encoding of the cluster labels, the value of each context variable can be determined by the values of all other context variables. For instance, consider the graph depicted in Figure 3.1. If we assume that the context variable C_1 captures interventions on the variable X_1, we would want to find the edge $C_1 \to X_1$ in our graph prediction. However, since the value of C_1 can be determined by the values of C_2 and C_3, a correct CI test would detect an independence between X_1 and C_1, conditioned on C_2 and C_3. This conditional independence is unfaithful with respect to the true causal graph and

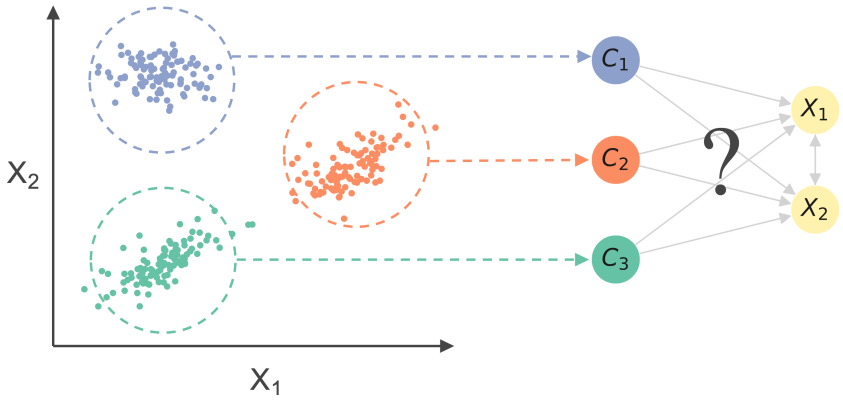


Fig. 3.1 Illustration of combining the PC-JCI algorithm with clustering. In the first step, the data is separated into clusters (left, here: 3 clusters in a 2-dimensional feature space). In the second step, the one-hot encoded cluster labels are incorporated into the causal search as context variables (right, here: 3 context variables and 2 original variables that correspond to the dimensions in the clustering space).

would result in an erroneous removal of the edge $C_1 \to X_1$. However, empirically this did not affect the overall prediction performance, as we mitigate this particular faithfulness violation by incorporating background knowledge into the prediction.

3.2.1 Incorporating background knowledge

Incorporating background knowledge on existing edges or gaps into the causal search can help to improve the prediction of the causal graph. For instance, in our setting, the context variables that emerge from clustering the data are root nodes in the graph, which means that they can not be caused by any other variables (as seen in Figure 3.1). This is equivalent to the JCI assumptions 1, 2, and the second part of assumption 3.

We pass this knowledge to the PC algorithm by removing all incoming edges to context variables. It follows that there are no undirected edges between the context variables in the skeleton of the predicted graph. In the PC algorithm, all edges between context variables are removed at the beginning of the skeleton prediction, when vertex-pairs are tested for marginal independencies. Since the context variables are removed from each other's neighbourhoods, they are not included in the potential separating sets for any vertex-pairs that involve a context variable. This significantly reduces the number of CI tests and hence the risk of erroneous testing results. Additionally, it prevents the prediction from being affected by the described faithfulness violations, as the algorithm does not test for the unfaithful independence between a context variable C_i and an endogenous variable X_j, conditioned on all other context variables $\mathcal{C}\backslash\{C_i\}$.

The background knowledge is also used when orienting the edge between a context variable and an endogenous variable. This is done for all detected undirected edges between context and endogenous variables before the second step of the PC algorithm, i.e. before unshielded colliders are identified and oriented. The background knowledge is given priority over unshielded colliders, which means that unshielded colliders are only oriented if they do not conflict with the incorporated background knowledge.

3.2.2 Limitations

We expect the success of PC-JCI with clustering to depend on a number of factors, such as the right choice of hyperparameters, properties of the observational and interventional distributions, and the convergence behaviour of the clustering algorithm. We discuss the most relevant limitations below.

Number of clusters

Many clustering algorithms, including k-means and GMM clustering, require the specification of the number of clusters as a hyperparameter. In practice, the number of data-generating SCMs may not be known. When considering only perfect interventions on single or multiple targets, a dataset with D variables could have been generated by up to 2^D possible SCMs, assuming that there is only one SCM for each set of intervention targets. Correspondingly, it is difficult to choose a suitable number of clusters in practice. While it might still be possible to improve causal discovery even with an incorrect number of clusters, it hinders the clustering algorithm from separating the data according to the true generating SCMs. Any clustering that differs from this separation increases the risk of introducing errors into the graph prediction.

Figure 3.2 shows an example of how a misspecified number of clusters $K > R+1$ can result in the prediction of a false negative edge. In this example, we have two causal variables X_1 and X_2 that are linearly dependent in the true data-generating SCM. We only assume one SCM here, i.e. $R = 0$ interventions. On this data, we are fitting k-means with $K = 2$ clusters, which produces the cluster label assignment as shown in Figure 3.2. When looking at the clusters separately, the dependency between both variables is significantly weaker. In this example, the dependency of X_1 and X_2, conditioned on the clustering, would not be large enough to reject the null hypothesis,

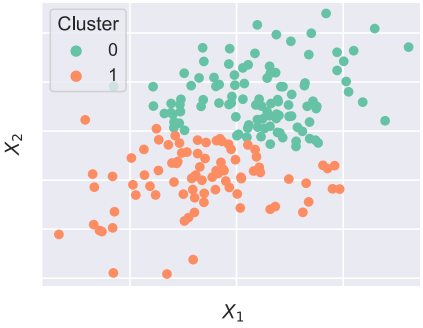


Fig. 3.2 Linearly dependent toy data with two variables, clustered with k-means.

given a commonly used threshold value like $\alpha = 0.01$. Since the clustering variable acts as a separating set for X_1 and X_2, this would result in the prediction of a false negative edge.

Separability in clustering space

A dataset that has been generated by SCMs with overlapping distributions is not completely separable by any function, including clustering. Even when the clustering returns the correct cluster characteristics, such as the centers for k-means or distributions and mixture coefficients for the GMM, the amount of correct assignments is bounded by the distribution overlap. Hence, the only attainable goal is to identify the characteristics of each SCM, instead of identifying every interventional datapoint correctly.

Depending on the data and clustering algorithm, there might exist alternative clusterings that are better suited with respect to the clustering objective. For example, if the data is generated by multimodal SCMs, it could be better to separate the data according to the modes within the same SCMs instead of separating it according to the generating SCMs. Generally, the observational and interventional distributions are difficult to separate if the distances in the clustering space within an SCM are large compared to the distances between SCMs.

One consequence of this is the introduction of spurious correlations if the clustering fails to separate the contexts. For instance, consider two SCMs with two independent endogenous variables $\mathcal{V} = \{X_1, X_2\}$, where both variables are shifted in the interventional SCM compared to the observational one. If the data is not separated according to the SCMs, the shift might appear as a dependence in the CI test, and hence the PC algorithm will predict a false positive edge. Note that the likelihood of a false positive edge increases even if only one variable is shifted in the interventional SCM, as the sampling noise might introduce a small shift in the other variables. Thus, the likelihood of a linear dependency occurring under the null hypothesis increases, and might be underestimated by a CI test like Fisher's z-transformation, which assumes a multivariate Gaussian distribution.

3.3 ALLIN

In Section 3.2, we have presented a straightforward combination of existing algorithms, *PC-JCI with clustering*, that sequentially identifies interventions and predicts the causal graph. While this approach offers a strong baseline for causal discovery from mixed datasets with latent interventions, we aim to investigate whether we can improve on this baseline with a joint approach. Additionally, our assumptions on interventions are not encoded in the intervention detection step of *PC-JCI with clustering*, such that it might be challenging for a clustering algorithm to separate the data according to the data-generating SCMs.

To improve upon the sequential approach, we propose the causal discovery method ALLIN (*Assigning Losses to Latent INterventions*). ALLIN alternates between learning the SCMs and assigning datapoints to interventional SCMs. For this, the model learns the observational SCM $\mathcal{M}^{(\varnothing)} \in \boldsymbol{\mathcal{M}}$ and restricts the set of possible SCMs $\boldsymbol{\mathcal{M}} \backslash \mathcal{M}^{(\varnothing)}$ to interventional versions of $\mathcal{M}^{(\varnothing)}$. We introduce the SCM learning step and the intervention detection step separately in Section 3.3.1 and 3.3.2 respectively. As each step relies on the estimate of the other step, we explain how the steps are connected in Section 3.3.3, including the initialization of the SCM estimate at the first step.

3.3.1 Learning the SCMs

As a starting point of ALLIN, we adapt the causal discovery algorithm NOTEARS (Zheng et al., 2018). While NOTEARS only learns the weights of the observational SCM, ALLIN aims to learn the observational SCM $\mathcal{M}^{(\varnothing)}$ and the interventional SCMs $\mathcal{M}^{(I_1)}, ..., \mathcal{M}^{(I_R)}$. Since we consider only perfect interventions on $\mathcal{M}^{(\varnothing)}$, which can occur on either one or multiple targets, we directly integrate this assumption into the objective of ALLIN. Additionally, we consider the possibility that the mean of an interventional distribution is shifted compared to the observational distribution. To model all SCMs including potential shifts, we extend the unconstrained loss function of NOTEARS such that ALLIN has two options for predicting each variable of a datapoint: it can choose between using the observational and interventional regime. Figure 3.3 shows an illustration of this underlying principle.

Like NOTEARS, the objective of ALLIN is subject to a constraint, which makes it necessary to formulate a dual objective for differentiable optimization. Below, we present the unconstrained loss of ALLIN and its dual modification used for constrained optimization.

Unconstrained loss

To include both regimes in the objective, the unconstrained loss $\ell(\mathbf{X})$ is a weighted sum of the losses of the observational regime $\ell_{obs}(\mathbf{x}_m)$ and the interventional regime $\ell_{int}(\mathbf{x}_m)$ for each datapoint index $\mathbf{x}_m \in \mathbf{X}$. For each datapoint $\mathbf{x}_m$, we weigh the feature-wise losses by the feature-wise probabilities $\mathbf{p}(\mathbf{x}_m) \in [0,1]^D$ and $\mathbf{1} - \mathbf{p}(\mathbf{x}_m)$ to be observational or interventional:

$$\ell(\mathbf{X}) = \frac{1}{2M} \sum_{m=1}^{M} \sum_{d=1}^{D} \ell_{obs,d}(\mathbf{x}_m) \cdot [\mathbf{p}(\mathbf{x}_m)]_d + \ell_{int,d}(\mathbf{x}_m) \cdot (1 - [\mathbf{p}(\mathbf{x}_m)]_d) \tag{3.2}$$

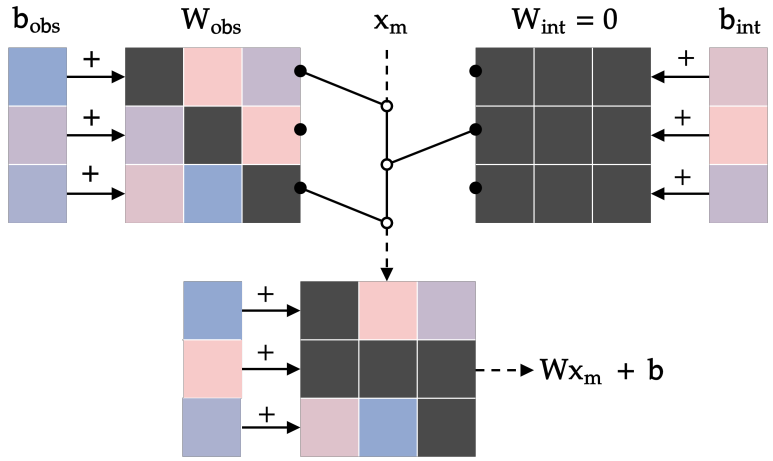


Fig. 3.3 Underlying assumption on the mixture of observational and interventional data. A datapoint $\mathbf{x}_m$ is assumed to be generated by a mixture of observations (left) and interventions (right) on the variables. For the interventional variables, all inputs are dropped out by replacing the corresponding rows of the observational weight matrix with zeros (here: second row). ALLIN aims to learn the observational SCMs ($\mathbf{W}_{\text{obs}}\mathbf{x}_m + \mathbf{b}_{\text{obs}}$), the interventional SCMs ($\mathbf{b}_{\text{int}}$) and the assignments of datapoints to these SCMs. The diagonal of $\mathbf{W}_{\text{obs}}$ is set to 0, such that variables are not predicted from themselves. Zero weights are depicted in gray, colours denote non-zero values.

For the main version of ALLIN, the overall loss is a weighted mixture of Mean Squared Errors (MSE), where the observational and interventional losses are calculated as follows:

$$\ell_{obs}(\mathbf{x}_m) = (\mathbf{x}_m - [\mathbf{W}_{obs}\mathbf{x}_m + \mathbf{b}_{obs}])^2 \tag{3.3}$$

$$\begin{aligned}\ell_{int}(\mathbf{x}_m) &= (\mathbf{x}_m - [\mathbf{W}_{int}\mathbf{x}_m + \mathbf{b}_{int}])^2 \\ &= (\mathbf{x}_m - \mathbf{b}_{int})^2\end{aligned} \tag{3.4}$$

Since the interventional regime assumes perfect interventions, it drops out all inputs to the prediction of a variable. In NOTEARS, the model only learns the weight matrix $\mathbf{W}_{obs}$ but no bias $\mathbf{b}_{obs}$. This is not necessary for NOTEARS because the data is commonly zero-centered before learning $\mathbf{W}_{obs}$, such that the mean of each variable is 0. With ALLIN, the data is also zero-centered and accordingly the mean of each variable in the mixture is also 0. However, when assigning the data to observational and interventional regimes, it is possible that the means of these individual regimes are not 0. For example, there could be an equal amount of observational and interventional datapoints with a mean of -1 and 1 respectively, resulting in an overall mean of 0. Because of this, the model allows for learning an observational and interventional bias $\mathbf{b}_{obs}$ and $\mathbf{b}_{int}$.

To encourage sparsity, a penalty is added, resulting in the following unconstrained loss:

$$F(\mathbf{W}_{obs}, \mathbf{b}_{obs}, \mathbf{b}_{int}) = \ell(\mathbf{W}_{obs}, \mathbf{b}_{obs}, \mathbf{b}_{int}; \mathbf{X}) + \lambda \|\mathbf{W}_{obs}\|_1 \tag{3.5}$$

The term penalizes the absolute values of the observational weights $\mathbf{W}_{obs}$ and is weighted with the hyperparameter λ.

Dual objective

Since $\mathbf{W}_{obs}$ determines the prediction of the causal graph, it is subject to the acyclicity constraint developed by Zheng et al. (2018), resulting in the following final objective:

$$\widehat{\mathbf{W}}_{obs}, \widehat{\mathbf{b}}_{obs}, \widehat{\mathbf{b}}_{int} = \min_{\substack{\mathbf{W}_{obs} \in \mathbb{R}^{D \times D} \\ \mathbf{b}_{obs} \in \mathbb{R}^{D} \\ \mathbf{b}_{int} \in \mathbb{R}^{D}}} F(\mathbf{W}_{obs}, \mathbf{b}_{obs}, \mathbf{b}_{int}) \qquad \text{subject to } h(\mathbf{W}_{obs}) = 0 \tag{3.6}$$

In line with NOTEARS, the objective used for optimization is a dual function with Lagrangian multipliers ρ and α:

$$D(\alpha) = \min_{\substack{\mathbf{W}_{obs} \in \mathbb{R}^{D \times D} \\ \mathbf{b}_{obs} \in \mathbb{R}^{D} \\ \mathbf{b}_{int} \in \mathbb{R}^{D}}} F(\mathbf{W}_{obs}, \mathbf{b}_{obs}, \mathbf{b}_{int}) + \frac{\rho}{2} |h(\mathbf{W}_{obs})|^2 + \alpha h(\mathbf{W}_{obs}) \tag{3.7}$$

3.3.2 Detecting interventions

In Equation 3.2, it is assumed that the assignment probabilities are known for each datapoint, but in practice, they must be estimated from the data. To stay within the differentiable framework, we learn a function $\mathbf{p}(\mathbf{x}_m)$ with a differentiable model $\mathbf{f}$. We optimize $\mathbf{f}$ to fit the data, applying an element-wise sigmoid function to its output, such that the predictions $\mathbf{p}(\mathbf{x}_m)$ are in the interval $(0, 1)$, and resemble probabilities:

$$\mathbf{p}(\mathbf{x}_m) : \mathbb{R}^D \to (0,1)^D : \mathbf{x}_m \mapsto \sigma(\mathbf{f}(\mathbf{x}_m)) \tag{3.8}$$

For the scope of this thesis, the model $\mathbf{f}$ is chosen to be a simple Multilayer Perceptron (MLP). However, it is possible to exchange this model arbitrarily, e.g. to a different architecture, if this should be more suitable for the task.

Single objective

In the intervention detection steps, we fix the weight matrix $\mathbf{W}_{obs}$ and the biases $\mathbf{b}_{obs}$ and $\mathbf{b}_{int}$ returned by the last iteration of the SCM learning step. Because of this, we do not need to optimize for acyclicity when we learn the assignments to the different regimes. For optimizing the model $\mathbf{f}$, it is therefore sufficient to use a single objective. The main version of ALLIN uses the MSE presented in Equations 3.2, 3.3 and 3.4 for learning the assignments, but different loss functions can also be used.

As an alternative to the MSE loss, we explore an objective based on negative log-likelihood (NLL). In this objective, we assign the datapoints to different SCMs by fitting a mixture model to the data, which optimizes the likelihood of the data under a mixture of distributions. This corresponds to modelling both the weights of endogenous variables and the exogenous variables in a structural equation, while the MSE only takes the weights of endogenous variables into account, i.e. the distance of a datapoint to the mean as predicted by its input variables. When using the NLL, the loss function is based on a weighted sum of the observational and interventional data likelihoods Q_{obs} and Q_{int} respectively:

$$\ell(\mathbf{X}) = -\frac{1}{2M}\sum_{m=1}^{M}\sum_{d=1}^{D}\log\left(Q_{obs,d}(\mathbf{x}_m)\cdot[\mathbf{p}(\mathbf{x}_m)]_d + Q_{int,d}(\mathbf{x}_m)\cdot(1-[\mathbf{p}(\mathbf{x}_m)]_d)\right) \tag{3.9}$$

To model the different SCMs with Gaussian exogenous terms, the likelihood of the data is given by Gaussian probability distributions $\mathcal{N}$ of observational and interventional variables:

$$Q_{obs}(\mathbf{x}_m) = \mathcal{N}(\mathbf{x}_m \mid \mu_{obs}(\mathbf{x_m}), \Sigma_{obs}) \tag{3.10}$$

$$Q_{int}(\mathbf{x}_m) = \mathcal{N}(\mathbf{x}_m \mid \mu_{int}(\mathbf{x_m}), \Sigma_{int}) \tag{3.11}$$

When optimizing the NLL, the means of a datapoint $\mathbf{x}_m$ are given by $\mu_{obs}(\mathbf{x}_m) = \mathbf{W}_{obs}\mathbf{x}_m + \mathbf{b}_{obs}$ and $\mu_{int}(\mathbf{x}_m) = \mathbf{b}_{int}$ respectively. The exogenous terms are assumed to be independent of the input $\mathbf{x}_m$ and of each other, corresponding to exogenous variables of a Markovian SCM. Accordingly, the covariance matrices are diagonal with $\text{diag}(\Sigma_{obs}) = \sigma^2_{obs}$ and $\text{diag}(\Sigma_{int}) = \sigma^2_{int}$. For given assignments $\mathbf{p}(\mathbf{x}_m)$, these variances can be learned from the data, e.g. by optimizing two D-dimensional biases without inputs. Learning the assignments with the NLL allows modelling observational and interventional distributions with varying variances, instead of pushing the model to separate the data into similarly shaped low-variance distributions.

ALLIN with clustering

As an alternative version of ALLIN, we can use clustering algorithms like k-means or GMM clustering as a basis for detecting interventions. For this, we first generate a clustering that ideally corresponds to different data-generating SCMs. Then, we use the cluster memberships $\mathbf{C}$ as input to the assignment model in the intervention detection step of ALLIN, instead of using the raw input $\mathbf{X}$. The probability function $\mathbf{p}(\mathbf{x}_m)$ is therefore a composite of some clustering function $\mathbf{g}(\mathbf{x}_m)$ and a mapping function $\mathbf{f}(\mathbf{c}_m)$ that assigns each cluster to an SCM:

$$\mathbf{p}(\mathbf{x}_m) : \mathbb{R}^D \to \{0,1\}^K \to (0,1)^D : \mathbf{x}_m \mapsto \mathbf{c}_m \mapsto \sigma\left(\mathbf{f}(\mathbf{c}_m)\right) \tag{3.12}$$

Since $\mathbf{c}_m \in \{0,1\}^K$ is a one-hot encoded vector of cluster memberships, the function $\mathbf{f}(\mathbf{c}_m)$ only needs to learn which cluster membership to assign to which SCM, i.e. it needs to identify the intervention targets. Due to the compression of the input to cluster memberships, a linear transformation $\mathbf{f}(\mathbf{c}_m) = \mathbf{V}\mathbf{c}_m + \mathbf{a}$ is powerful enough to express all possible assignment functions. This compression also restricts the size of the set of identified SCMs to a maximum of K, as there is no function that can assign datapoints from the same cluster to different SCMs.

3.3.3 Algorithm

Algorithm 1 shows an outline of the ALLIN algorithm. At first, the observational weight matrix $\mathbf{W}_{obs}$ is estimated as if all datapoints were observational. This pretraining practically provides initial values for the set of SCMs. By definition, the pretraining is identical with the NOTEARS algorithm. Using the initialized SCMs, ALLIN alternates between learning the assignment probabilities $\mathbf{p}$ and learning the SCMs.

When searching for interventions, ALLIN fixes the SCMs, i.e. the weights and biases $\mathbf{W}_{obs}$, $\mathbf{b}_{obs}$ and $\mathbf{b}_{int}$. Since the dual objective tolerates a certain value of the cyclicity measure $h(\mathbf{W}_{obs})$, the weight matrix $\mathbf{W}_{obs}$ can still include traces of cyclicity, such as weights for predicting causes from effects. Such weights complicate the detection of interventions, as they can contain predictive power even for interventional datapoints, making it less beneficial to drop out all inputs. Thus, we threshold the weights $\widetilde{\mathbf{W}}_{obs}$ with the weight threshold ω to remove these traces:

$$\widetilde{W}_{ij} = \begin{cases} W_{ij} & \text{if } W_{ij} \geq \omega \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

For learning the SCMs, ALLIN fixes the updated model $\mathbf{f}$ and optimizes the dual objective. Before this, the SCMs are reinitialized, such that it is easier for the model to establish a new causal order influenced by the detected interventions.

Algorithm 1: ALLIN

Input: Data $\mathbf{X}$, sparsity penalty λ, weight threshold ω
Output: Adjacency matrix $\mathbf{A}$
$\mathbf{W}_{obs} \leftarrow \mathbf{0}$;
$\mathbf{b}_{obs} \leftarrow \mathbf{0}$;
$\mathbf{b}_{int} \leftarrow \mathbf{0}$;
`// Pretraining`
$\mathbf{p} \leftarrow \mathbf{1}$;
$\mathbf{W}_{obs} \leftarrow \text{argmin } F(\mathbf{W}_{obs}; \mathbf{X}, \mathbf{p}, \mathbf{b}_{obs}, \mathbf{b}_{int}, \lambda)$ subject to $h(\mathbf{W}_{obs}) = 0$;
for $i \leftarrow 0$ **to** T **do**
    `// Detecting interventions`
    $\widetilde{\mathbf{W}}_{obs} \leftarrow (\mathbf{W}_{obs} \geq \omega) \odot \mathbf{W}_{obs}$;
    $\mathbf{f} \leftarrow \text{argmin } \ell\left(\mathbf{f}; \mathbf{X}, \widetilde{\mathbf{W}}_{obs}, \mathbf{b}_{obs}, \mathbf{b}_{int}\right)$;
    `// Learning the SCMs`
    $\mathbf{W}_{obs} \leftarrow \mathbf{0}$;
    $\mathbf{b}_{obs} \leftarrow \mathbf{0}$;
    $\mathbf{b}_{int} \leftarrow \mathbf{0}$;
    $\mathbf{W}_{obs}, \mathbf{b}_{obs}, \mathbf{b}_{int} \leftarrow \text{argmin } F(\mathbf{W}_{obs}, \mathbf{b}_{obs}, \mathbf{b}_{int}; \mathbf{X}, \mathbf{f}, \lambda)$ subject to $h(\mathbf{W}_{obs}) = 0$;
end
$\mathbf{A} \leftarrow \mathbf{W}_{obs} \geq \omega$;
return $\mathbf{A}$;

After a fixed amount of iterations, ALLIN returns its prediction of the adjacency matrix of the causal graph. In line with NOTEARS, ALLIN predicts an edge from variable X_i to X_j if the corresponding element W_{ij} of the weight matrix is at least as large as the threshold ω (see Equation 2.12).

3.3.4 Limitations

Due to the proximity of the methods, ALLIN is subject to the same limitations as NOTEARS. For example, ALLIN can be expected to converge to stationary points instead of a global optimum, since its loss landscape is nonconvex. Furthermore, it is developed for linear dependencies with additive Gaussian noise, and might not generalize well when this assumption is not satisfied. However, generalizations of NOTEARS to semiparametric and nonparametric SCMs exist (Zheng et al., 2020), and can likely be transferred to ALLIN.

Another known weakness of NOTEARS is its invariance to different scalings of the marginal distributions (Kaiser and Sipos, 2021; Reisach et al., 2021), which makes it imperative to normalize the data. It is also sensitive to the choice of the hyperparameters ω and λ which are both adopted by ALLIN. Beyond these known limitations of NOTEARS, there are limitations that apply specifically to ALLIN.

Detection of interventions on root nodes

For variables that are root nodes in the causal graph, the model does not benefit from dropping out the inputs when predicting interventional datapoints, as the variable does not have any parents in the true causal graph. Therefore, the edge-removing property of perfect interventions can not be leveraged to identify interventions on root nodes. Conversely, it is also not beneficial for the model to assign observational datapoint features to the observational regime. In fact, both regimes are initially identical when the observational and interventional biases are both zero. As a consequence of this, the assignments for root nodes are dominated by their random initializations. Since this initialization introduces noise into the means of the datapoints of both regimes, we can expect ALLIN to pick up on this noise and split the data variables according to their random biases.

Error propagation

Since the estimation of the SCMs, i.e. of $\mathbf{W}_{obs}$, $\mathbf{b}_{obs}$ and $\mathbf{b}_{int}$, provides the basis for detecting interventions, errors in this estimation can be propagated into the next iteration of learning the assignments. Vice versa, faulty assignments can have a negative effect on the estimations of the SCMs if their influence outweighs the correct assignments. Therefore, an error in one of these steps might have a cascading effect in the following steps. As part of our experiments in the following chapter, we will investigate the impact of these limitations.

4 Experiments

We have proposed two methods for the problem of causal discovery from a mixed dataset of observational and interventional distributions with latent interventions. In this chapter, we provide an examination of our methods through experimentation on linear Gaussian data. We introduce the specific setup of our experiments in Section 4.1. In Section 4.2, we present the outcomes of our experiments, including a comprehensive analysis of our methods.

4.1 Setup

In Section 4.1.1, we describe how we generate causal graphs, and then sample mixed observational and interventional datasets from these graphs. In our experiments, we compare six different methods. In Section 4.1.2, we cover the setup of these methods. In Section 4.1.3, we introduce the metrics we employ to evaluate the methods' performance.

4.1.1 Data generation

In each experiment, we generate a random graph $\mathcal{G}$ with $D = 10$ variables $\mathcal{V} = \{X_1, ..., X_{10}\}$ following the Erdős-Rényi model by sampling edges with an edge probability p. The Erdős-Rényi model creates all graphs with 10 vertices and a fixed number of edges with equal probability, and is commonly used for the generation of causal graphs to evaluate causal discovery methods, e.g. in Zheng et al. (2018), Brouillard et al. (2020) and Faria et al. (2022). While a higher number of variables reveals performance differences between methods more clearly, we restrict our experiments to graphs with 10 variables, which is in line with Brouillard et al. (2020) and Faria et al. (2022). This enables us to perform a larger amount of experiments within the time constraints of this thesis, and run each experiment on 50 different causal graphs, accounting for the relatively high performance variance that results from the variance in the graph generation. By imposing a causal order $(X_1, ..., X_{10})$ on the graph and generating only edges $X_i \to X_j$ for which $i < j$, the generated graphs are ensured to be DAGs.

Causal discovery methods usually make more errors when the number of edges in the true causal graph is higher. Additionally, a higher number of edges means that a perfect intervention is expected to cause a larger change in the graph, as more incoming edges to a vertex are removed. In order to investigate how different numbers of edges affects causal discovery in our setting, we study three different types of Erdős-Rényi graphs with varying connectivities. The connectivity of a graph can be described by the average neighbourhood size N of each node, i.e. by the average amount of adjacencies that each node has. Since the graphs are randomly generated, N can not directly be controlled for each graph. However, the expected neighbourhood size $\mathbb{E}[N]$ over all graphs can be controlled by choosing the following edge probability:

$$p = \frac{\mathbb{E}[N]}{D} \tag{4.1}$$

In our experiments, we generate graphs with the expected neighbourhood sizes $\mathbb{E}[N] \in \{2,3,4\}$ through sampling edges with probability $p \in \{0.2,0.3,0.4\}$. The expected number of edges in $\mathcal{G}$ is:

$$\mathbb{E}[|\mathcal{E}|] = \binom{D}{2} p \tag{4.2}$$

For our connectivity settings, this translates to $\mathbb{E}[|\mathcal{E}|] \in \{9, 13.5, 18\}$. The maximum number of edges in DAGs with 10 variables is 45. This means that we focus on relatively sparse graphs. Since sparseness reduces the number of necessary CI tests in constraint-based methods, it is a commonly assumed property in causal discovery methods (Kalisch and Bühlmann, 2007).

From a given graph $\mathcal{G}$, we generate a family of SCMs $\boldsymbol{\mathcal{M}}$. This family consists of one observational SCM $\mathcal{M}^{(\varnothing)} = (\mathcal{V},\mathcal{U},\mathcal{F})$ and $R = 10$ interventional SCMs $\mathcal{M}^{(I_1)},...,\mathcal{M}^{(I_{10})}$. The graph $\mathcal{G}$ is the graph representation of the observational SCM $\mathcal{M}^{(\varnothing)}$:

$$\mathcal{V} = \{X_1,...,X_{10}\} \qquad \mathcal{U} = \{U_1,...,U_{10}\} \qquad \mathcal{F} = \{f_1,...,f_{10}\} \tag{4.3}$$

$$f_i : X_i = U_i + \sum_{X \in Pa_i} X \tag{4.4}$$

The observational SCM $\mathcal{M}^{(\varnothing)}$ is a linear Gaussian system, where each variable X_i is a sum of its parents Pa_i, each weighted with a value of 1. The exogenous variables are Gaussian

distributed with mean 0 and standard deviation 0.5 for all $U \in \mathcal{U}$:

$$U_i \sim \mathcal{N}\left(0, 0.5^2\right) \tag{4.5}$$

Meanwhile, each interventional SCM $\mathcal{M}^{(I_r)} = (\mathcal{V}, \mathcal{U}, \mathcal{F}^{(I_r)})$ describes the system under a perfect intervention on variable X_r, i.e. the type of intervention that ALLIN is designed to detect. We define each interventional SCM $\mathcal{M}^{(I_r)}$ as follows:

$$\mathcal{V} = \{X_1, ..., X_{10}\} \qquad \mathcal{U} = \{U_1, ..., U_{10}\} \qquad \mathcal{F} = \{f_1^{(I_r)}, ..., f_{10}^{(I_r)}\} \tag{4.6}$$

$$f_i^{(I_r)} : X_i = \begin{cases} U_i + 1 & \text{if } i = r \\ U_i + \sum_{X \in Pa_i} X & \text{otherwise} \end{cases} \tag{4.7}$$

Only the variable X_i with $i = r$ is an intervention target of the SCM $\mathcal{M}^{(I_r)}$, such that we only have single-target interventions. For the other variables in an interventional SCM, the structural equations are the same as in the observational SCM. Unlike in the observational structural equation, the mean of the corresponding interventional distribution is shifted by a value of 1, resulting in an *interventional mean shift*. We shift the mean, as we expect that our baseline *PC-JCI with clustering* might require a mean difference to separate the data according to the generating SCMs. For our method ALLIN, we expect that it is able to model interventional mean shifts without requiring them.

From each SCM, we sample 2000 datapoints, such that the size of the dataset in an experiment is $M = 11 \cdot 2000 = 22000$. As input to the clustering algorithm, we use the unnormalized data, whereas for the causal discovery algorithms, the data is normalized before being processed. This is important for the differential method NOTEARS and our method ALLIN, as these methods can otherwise simply infer the causal order of the graph by sorting the variables according to their marginal variances (Reisach et al., 2021). However, clustering the data with a Gaussian mixture model (GMM), which is the main clustering algorithm used in our experiments, is not affected by the scale of the marginal variances.

4.1.2 Methods

We compare our baseline PC-JCI with clustering and our method ALLIN to the normal PC algorithm (Spirtes et al., 1993) and NOTEARS (Zheng et al., 2018). Additionally, we run experiments with a differential causal discovery method developed for the setting of mixed datasets with latent interventions (Faria et al., 2022), which we will call DCDI-L. Finally, we

use ENCO (Lippe et al., 2022) as a reference method for the setting of known interventions with known targets. The methods are specified and implemented as follows.

PC algorithm. We use the stable version (Colombo et al., 2014) of the PC algorithm and resolve conflicts between unshielded colliders with the help of additional CI tests, resulting in order-independent graph predictions. The algorithm is implemented with the causal-learn package (CMU-CLeaR, 2021). We modified this implementation, such that invalid CI tests return nan-values. To match the behaviour of causal-learn with the pcalg package (Kalisch et al., 2012), CI tests with nan-values result in the removal of the corresponding edge. As a CI test, we use Fisher's z-transformation with a threshold of $\alpha = 0.01$.

PC-JCI with clustering. The settings of the causal discovery part of the algorithm are identical with the ones for the original PC algorithm. For clustering, we use the scikit-learn implementations of GMM and k-means (Pedregosa et al., 2011). We call this baseline method *PC-JCI GMM* or *PC-JCI k-means* respectively. In both cases, the number of clusters is set to the size of $\mathcal{M}$, i.e. to the true number of data-generating SCMs. For our main settings, this is $R+1 = 11$.

NOTEARS. We use the implementation provided by the authors[1] with the MSE as described in Section 2.2.3. In line with Zheng et al. (2018), we optimize the unconstrained loss with the L-BFGS-B optimizer for each inner step of the dual function. The hyperparameters λ and ω are tuned based on two connectivity settings $\mathbb{E}[N] \in \{2,4\}$. The seeds used for tuning differ from the seeds used in our experiments, which means that the parameters were tuned on different graphs. For a detailed description of the tuning process, see Appendix A. The resulting values are $\lambda = 0.22$ and $\omega = 0.03$, which we found to be roughly optimal for graphs with $\mathbb{E}[N] = 2$.

ALLIN. To learn the SCMs, we optimize the dual function of ALLIN similarly to NOTEARS, using the L-BFGS-B optimizer for the unconstrained loss. For the hyperparameters λ and ω, we use the same values as for NOTEARS. The assignment learning model is a MLP with 3 hidden layers of 128, 64 and 32 units, which is implemented in PyTorch (Paszke et al., 2017). It is optimized with the Adam optimizer, a learning rate of 1e-3 and a batch size of 128. We distinguish between the following 3 versions of ALLIN: (1) ALLIN with a loss based on the MSE (*ALLIN*), (2) ALLIN with a loss based on the NLL (*ALLIN Gaussian*) and (3) ALLIN with GMM clustering (*ALLIN GMM*).

[1] https://github.com/xunzheng/notears

DCDI-L. For this method, we use the public code[2] provided by the authors. Except for the truncation value K, we use the default values as given in the code. For K, we use the value found to be optimal in Faria et al. (2022) in a similar setting, which is identical with the number of SCMs, i.e. $K = 11$.

ENCO. We use the publicly available code[3] provided by Lippe et al. (2022), without changing any of the hyperparameters. In line with our setting, we use the version of ENCO that assumes causally sufficient graphs.

4.1.3 Metrics

To evaluate the quality of the prediction of a causal graph, we measure the Structural Hamming Distance (SHD) between the prediction and the true causal graph. If both graphs are DAGs, the SHD is the amount of edges that need to be added (False Negatives, FN), removed (False Positives, FP) or flipped in their causal direction (Flips) to arrive at the true causal graph:

$$SHD = FN + FP + Flips \tag{4.8}$$

The PC algorithm can predict mixed graphs with undirected edges. An undirected edge adds a value of 1 to the SHD, either because it is completely missing in the true causal graph or because it is oriented in one direction.

The explicit or implicit clustering performed by our methods is evaluated with the Adjusted Rand Index (ARI) against the ground truth clustering where datapoints are separated according to their generating SCMs. Given a dataset of M elements $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_M\}$, a predicted clustering $\mathcal{Y} = \{Y_1, ..., Y_C\}$ consisting of C clusters, and a ground truth clustering $\mathcal{Z} = \{Z_1, ..., Z_K\}$ with K clusters, the RI can be calculated from the number of agreements between $\mathcal{Y}$ and $\mathcal{Z}$:

$$RI = \frac{a+b}{\binom{M}{2}} \tag{4.9}$$

Here, a denotes the number of pairs of elements in $\mathbf{X}$ that are in the same cluster in $\mathcal{Z}$ as they are in $\mathcal{Y}$. The number of pairs of elements that are in different clusters in both clusterings are denoted with b. The RI lies in the interval $[0, 1]$, with 0 indicating that the clusterings disagree on all pairs of datapoints, and with 1 indicating that the clusterings are identical apart from permutations of the cluster order. The RI is adjusted for chance by subtracting the expected RI

[2] https://github.com/goncalorafaria/causaldiscovery-latent-interventions
[3] https://github.com/phlippe/ENCO

under random clusters $\mathbb{E}[RI]$ and rescaling:

$$ARI = \frac{RI - \mathbb{E}[RI]}{\max(RI) - \mathbb{E}[RI]} \tag{4.10}$$

Through this adjustment, the ARI is 0 when the clustering is as good as a random clustering. Its maximal value is 1, and it can take on negative values if the clustering is worse than random.

4.2 Results

In this section, we present the results of our experiments. We start by exploring different versions of ALLIN in Section 4.2.1, which we use to choose the best-performing version for further experiments. We compare this version against our baseline PC-JCI GMM and the existing methods NOTEARS, PC and DCDI-L in Section 4.2.2.

4.2.1 Different versions of ALLIN

In Section 3.3, we present our method ALLIN, which uses the MSE loss to learn the SCMs and identify interventions. As an alternative loss for identifying interventions, we describe a loss function that is based on NLL, which we use for the version *ALLIN Gaussian*. Additionally, we experiment with one version of ALLIN (MSE) which uses the cluster memberships returned by a GMM as inputs to the model $\mathbf{f}$ that learns the assignments (*ALLIN GMM*). Figure 4.1 shows the SHD of ALLIN, ALLIN Gaussian and ALLIN GMM after each iteration of the algorithm on datasets created from graphs of different connectivities $\mathbb{E}[N] \in \{2,3,4\}$, as described in Section 4.1.1. For zero iterations, the methods are by definition equal to NOTEARS, such that their performances are the same. The figure shows that ALLIN outperforms NOTEARS on all settings when choosing the optimal number of iterations.

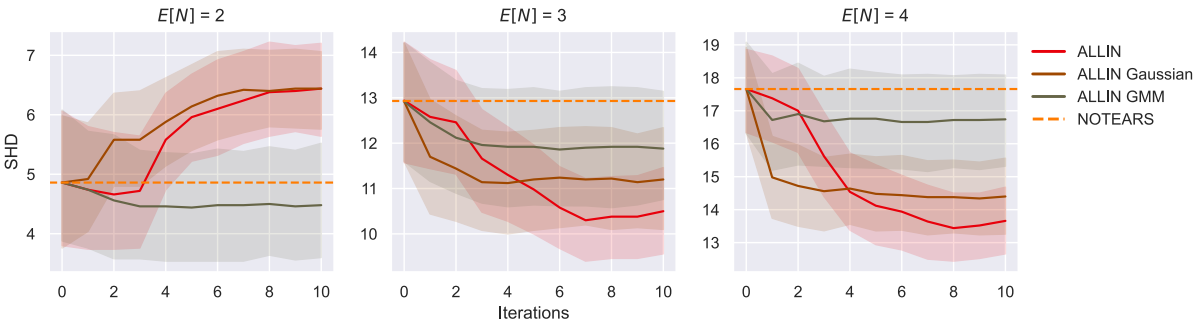


Fig. 4.1 SHD between the true graphs and the predicted graphs; results of 50 random seeds with a confidence interval of 95%. Predictions are made by ALLIN, ALLIN Gaussian and ALLIN GMM. As a reference, the performance of NOTEARS is added to the figure.

Optimal number of iterations. Table 4.1 shows the optimal number of iterations for ALLIN on each connectivity setting and the corresponding performance. For the higher-connectivity settings $\mathbb{E}[N] \in \{3,4\}$, ALLIN converges to optimal performance after 7-8 iterations. For the low-connectivity setting $\mathbb{E}[N] = 2$, ALLIN converges to a suboptimal performance worse than NOTEARS. In this setting, it reaches optimal performance after two iterations. For all following experiments, the number of iterations is chosen according to Table 4.1, unless stated otherwise.

Table 4.1 Optimal number of iterations for ALLIN.

$\mathbb{E}[N]$	Iterations	SHD
2	2	4.66 ($\pm$ 3.60)
3	7	10.30 ($\pm$ 3.22)
4	8	13.44 ($\pm$ 3.78)

Low connectivity. It is noticeable that the performance of ALLIN suffers under more iterations on low-connectivity graphs with $\mathbb{E}[N] = 2$. A possible reason for this is that ALLIN assigns too many datapoints to the interventional regime, such that the remaining signal for existing edges is too weak to overcome the resistance of the sparsity penalty and acyclicity constraint. Figure 4.2 provides an indication that this is indeed the case: when ALLIN assigns more than about 25% of all data to the interventional regime, performance begins to degrade. In the low-connectivity setting, this happens already after 2 iterations, whereas for $\mathbb{E}[N] \in \{3,4\}$, it takes 7 or 9 iterations respectively. The number of iterations until 25% interventional assignments coincides almost perfectly with the optimal number of iterations for each connectivity setting. As a reference, the true share of interventions is $1/11$, as we have one interventional SCM with 0% interventions and 10 interventional SCMs with 10% interventions (interventions on 1 out of 10 variables). Hence, the share of interventional assigments is almost 3 times the true share of interventions when performance starts to degrade.

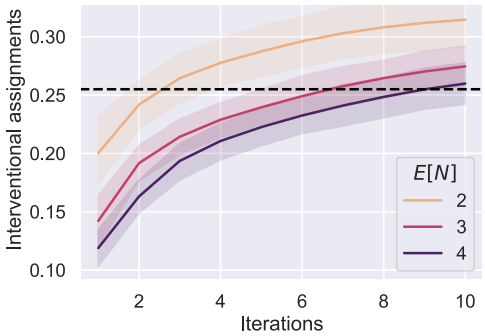
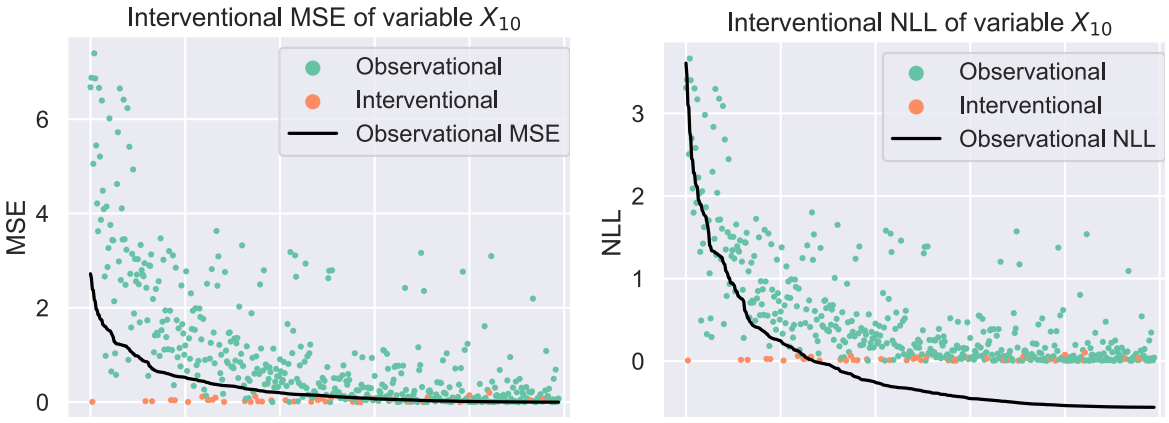


Fig. 4.2 Share of interventional assignments by ALLIN. Dashed line shows the value that coincides with the optimal number of iterations.

MSE versus NLL. On all connectivity settings, ALLIN outperforms ALLIN Gaussian when the number of iterations is optimal. For our data setup, using the MSE seems to be superior to using the NLL when learning the assignments of datapoints to the observational or interventional regimes.

Figure 4.3 illustrates an example of this. Based on the NOTEARS prediction of the weights, ALLIN learns the assignments to the regimes. Figure 4.3a and 4.3b show the losses of variable X_{10} under the interventional regime at the start of learning the assignments, using the MSE- or NLL-based loss respectively with the predicted NOTEARS weights. For all datapoints that lie beneath the black line, it would be beneficial for the model to assign these to the interventional regime.



(a) MSE. (b) NLL.

Fig. 4.3 Comparison of the MSE and NLL on variable X_{10} of all datapoints from one example run. The datapoints are sorted in descending order of their observational loss (x-axis) and their interventional loss is displayed on the y-axis. Datapoints generated by an intervention on X_{10} are colored orange.

This is just a snapshot of the losses at the very start of the assignment learning. For the NLL, the observational and interventional losses will change even during the first assignment learning step already, as the model will adjust its estimation of the variances of both distributions. At the start, about half the datapoints are randomly assigned to the interventional regime, such that the variance of interventional datapoints is initialized with a value of about $\sigma^2_{int,i} = 1$ for each variable index $i \in \{1,...,D\}$, resulting from the normalization of the data. Due to this variance, the loss has a minimum value of 0 under the interventional regime, as the constant term of the loss function is omitted:

$$\min \ell_{int,i}(\mathbf{x}_n) = \frac{1}{2}\left(\ln(\sigma^2_{int,i})\right) = 0 \tag{4.11}$$

Hence, there are no datapoints with an interventional NLL smaller than 0. For the observational NLL, a large part of the marginal variance can be explained through the input variables, resulting in a lower conditional variance and therefore in values smaller than 0.

Under the MSE, more datapoints that have both small observational and interventional losses are mistakenly assigned to the interventional regime, i.e. datapoints that suit both regimes well. When using the NLL, more observational datapoints that have a high loss in both regimes would mistakenly be assigned to the interventional regime in this example. Overall, the MSE seems to be more suitable to separate the decisive datapoints.

Table 4.2 Clustering performance and interventional assignments of ALLIN GMM after 10 iterations.

$\mathbb{E}[N]$	GMM ARI	ALLIN GMM ARI	Interventional assignments
2	0.234	0.219	21.80%
3	0.252	0.205	14.20%
4	0.248	0.192	9.46%

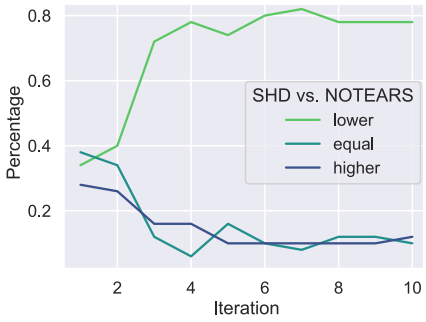
ALLIN GMM. Figure 4.1 shows that ALLIN GMM converges in about 4 iterations, reaching a performance that improves over NOTEARS on all connectivity settings. However, its improvement is limited to an SHD difference of about 1 compared to NOTEARS, such that it falls short of the other versions of ALLIN for graphs with $\mathbb{E}[N] \in \{3,4\}$. To understand the behaviour of ALLIN GMM, we show its clustering performance and the share of interventional assignments learned by the model in Table 4.2. The amount of interventional assignments is significantly lower for this version of ALLIN and does not reach 25%, which is approximately the amount of interventional assignments that coincides with a decrease in performance for the main version of ALLIN. This could explain why ALLIN GMM outperforms the other versions on low-connectivity graphs, as it might not experience the signal loss issues that are suspected to impact the main version of ALLIN.

Using the cluster memberships as inputs for learning the assignments seems to result in stricter requirements for assigning datapoints to the interventional regime. The model can only assign a full cluster to an interventional SCM, which may not be beneficial for the model, as the clusters might not be separated well and the model has not yet learned the bias terms of the individual regimes. Table 4.2 shows the clustering quality of the GMM and the quality of the implicit clustering that ALLIN GMM performs through assigning the clusters to SCMs. The ARI of ALLIN GMM appears to be bounded by the ARI of the GMM, which is better than random clustering, but only moderately similar to the ground truth clusters. Theoretically, the clustering could improve if ALLIN GMM was to join clusters that were mistakenly separated by the GMM. However, it is not able to split the clusters provided by the GMM further. Since ALLIN GMM seems to suffer from the GMM's limitations without exhibiting the potential for large improvements over NOTEARS, we will not include it in our further experiments.

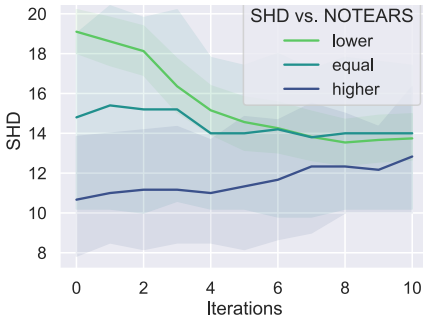
Error propagation. To investigate whether errors introduced by ALLIN have a cascading effect over iterations resulting in further errors, we separate the runs into three categories, according to the SHD difference between NOTEARS and ALLIN: (1) runs with an SHD lower than NOTEARS (improvement), (2) an SHD equal as NOTEARS, and (3) an SHD higher than NOTEARS (worsening). This way, we can isolate cases where ALLIN has a detrimental effect on the performance of causal discovery.

Figure 4.4a shows how the shares of these three categories change depending on the iteration of ALLIN, on graphs with $\mathbb{E}[N] = 4$. With more iterations of ALLIN, the percentage of runs that improve over NOTEARS increases up to 80%, such that the improvement cases make up a large majority of the runs. After 1 iteration, ALLIN worsens the performance of NOTEARS in about 30% of the runs. In these cases, ALLIN introduces additional error into the causal discovery process, compared to NOTEARS. However, in most cases, this error is not propagated. ALLIN is able to recover in about $2/3$ of these cases in later iterations, such that the percentage of worsening is only at about 10% after 5 iterations.

Figure 4.4b shows the SHD of the runs where the performance of ALLIN is worse than NOTEARS after 10 iterations, i.e. for runs where ALLIN is not able to recover from the introduced errors. For these runs, the prediction of ALLIN gets increasingly worse with a higher number of iterations. This suggests that in a few cases, error propagation between the iterations might indeed have a cascading effect on the performance of ALLIN.



(a) Percentage.



(b) SHD.

Fig. 4.4 Runs with lower, equal and higher SHD than NOTEARS, depending on the iteration.

4.2.2 Comparison of ALLIN, PC-JCI GMM and existing methods

Figure 4.5 shows the performance of ALLIN and PC-JCI GMM compared to NOTEARS, the PC algorithm and DCDI-L. In the low-connectivity setting with $\mathbb{E}[N] = 2$, the methods ALLIN, NOTEARS and PC-JCI GMM yield a mean SHD of around 4.5 and significantly outperform the PC algorithm and DCDI-L. For higher connectivities $\mathbb{E}[N] \in \{3,4\}$, ALLIN and PC-JCI GMM reach a mean SHD of around 10 and 12-13 respectively. Overall, none of the existing methods outperform ALLIN significantly in any of the settings. Furthermore, ALLIN is superior to every method apart from PC-JCI GMM in at least one setting.

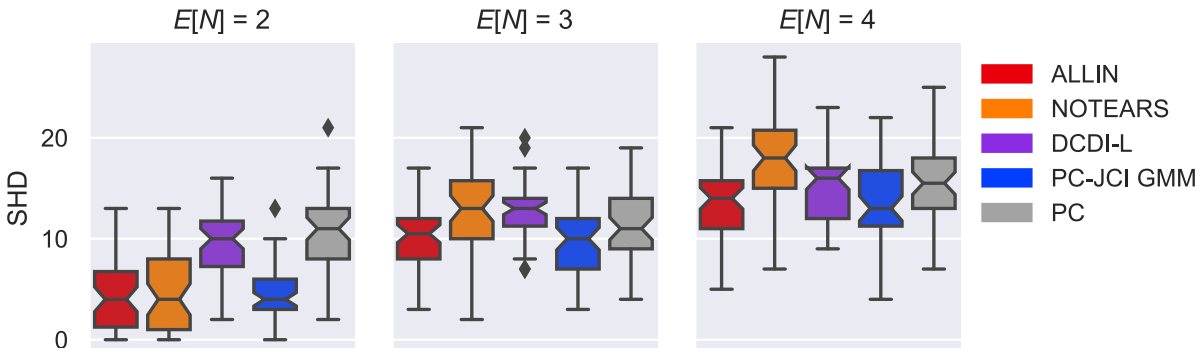


Fig. 4.5 SHD between the true graphs and the predicted graphs; results of 50 random seeds with a confidence interval of 95%. Predictions are made by the algorithms ALLIN, NOTEARS, DCDI-L, PC-JCI GMM and PC.

Runtime. Table 4.3 shows the runtime of the methods in seconds, averaged over all 150 experiments, i.e. including different connectivity settings $\mathbb{E}[N] \in \{2,3,4\}$. The fastest methods are PC and PC-JCI GMM which take on average less than a minute to predict the causal graph. However, the PC algorithm is known to scale badly to a higher number of variables, as it has a worst-case time complexity of $\mathcal{O}\left(D^{N_{\max}+2}\right)$, where $N_{\max}$ is the maximal neighbourhood size of any vertex in the graph. Since $N_{\max}$ can be expected to grow with D ($N_{\max} \leq D-1$), the PC algorithm has exponential time complexity in its worst-case. The GMM introduces even more variables into the graph prediction, such that PC-JCI GMM scales worse than PC. While the PC algorithm might eventually become slower than ALLIN as the number of variables grows, it is currently about 67 times faster than ALLIN for the generated graphs with 10 variables.

NOTEARS needs on average about 8-9 minutes. Due to the additional assignment learning step and the iterative approach, the runtime of ALLIN is longer with approximately 13 minutes on average. The method with the longest runtime is the DCDI-L, which takes on average 9-10 hours to complete a single run. Since DCDI-L does not provide results that are sufficiently compelling to justify its prolonged runtime, we do not include the method in our ablation studies for the scope of this thesis.

Table 4.3 Runtime of ALLIN, NOTEARS, DCDI-L, PC-JCI GMM and PC.

Method	**Runtime (s)**
ALLIN	776.60 ($\pm$ 545.34)
NOTEARS	506.34 ($\pm$ 505.08)
DCDI-L	35248.49 ($\pm$ 1888.29)
PC-JCI GMM	21.38 ($\pm$ 2.88)
PC	11.55 ($\pm$ 0.69)

4.3 Ablation studies

Both ALLIN and PC-JCI with clustering improve over existing methods on datasets generated from a mixture of observational and interventional SCMs. However, in our setting, we assume that single-target interventions on all variables are present and their means are shifted compared to observations. We conduct two ablation studies where we apply the methods to different datasets. In Section 4.3.1, the datasets were generated by fewer SCMs, i.e. we investigate datasets with interventions on fewer variables. We also use this setting to explore the potential of both methods by conducting experiments with oracle assignments or oracle clustering respectively. In Section 4.3.2, we explore how relevant the interventional mean shift is for the success of our methods.

Furthermore, we conduct ablation studies aiming to test the limitations of ALLIN and PC-JCI with clustering. In Section 4.3.3, we investigate the role of root nodes for the performance of ALLIN. In Section 4.3.4, we test PC-JCI with clustering and ALLIN under mismatches between the model specification and the data.

4.3.1 Interventions on fewer variables

ALLIN and PC-JCI GMM aim to exploit the presence of latent interventions in data. In practice, it is not known whether interventional data is actually present in a given dataset. To judge how these methods perform when fewer interventions are present, they are compared on datasets with a varying number of intervened variables. The experiments cover settings for $R \in \{0, 2, 4, ..., 10\}$ intervened variables, including the setting with interventions on each variable, as well as the purely observational setting when no interventional data is present. For all settings, the total number of samples is kept constant, resulting in fewer samples per SCM $\mathcal{M} \in \boldsymbol{\mathcal{M}}$ for datasets with more intervened variables.

Figure 4.6 shows the performance of ALLIN and PC-JCI GMM, depending on the number of intervened variables present in the data, i.e. for different sizes of $\boldsymbol{\mathcal{M}}$. These methods are compared to NOTEARS and PC, which do not exploit the presence of latent interventions. The performance of the methods shows a similar pattern for higher connectivities $\mathbb{E}[N] \in \{3, 4\}$: both ALLIN and NOTEARS improve with the number of intervened variables, and ALLIN has a consistent advantage over NOTEARS that increases with more intervened variables. Both PC methods are significantly superior compared to ALLIN and NOTEARS when no interventions are present, but the performance of standard PC decreases when the number of intervened variables is higher. In comparison, the performance of PC-JCI GMM stays approximately constant, independent of the number of intervened variables.

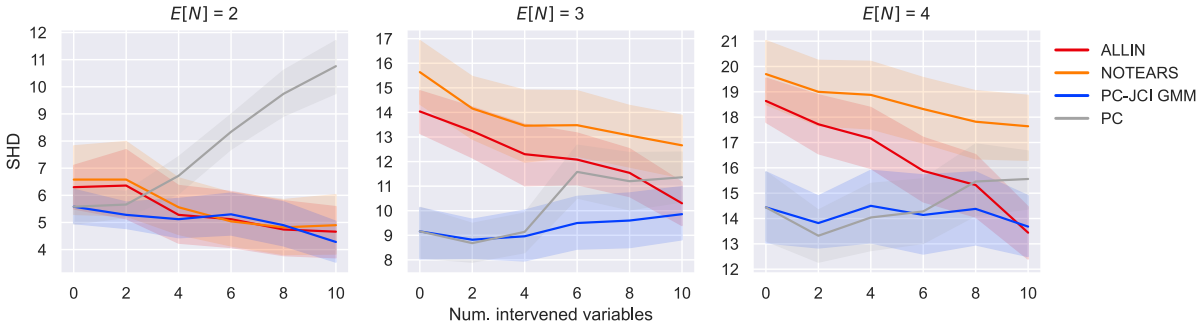


Fig. 4.6 SHD between the true graphs and the predicted graphs, depending on the number of intervened variables; results of 50 random seeds with a confidence interval of 95%. Predictions are made by ALLIN, NOTEARS, PC-JCI GMM and PC.

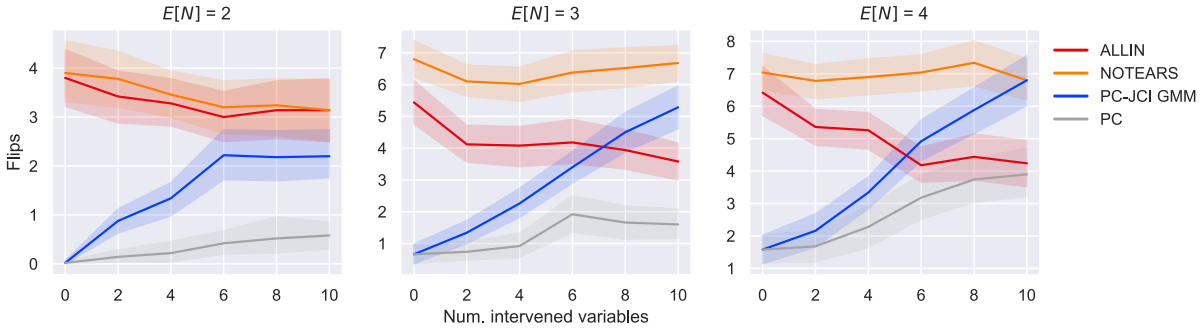


Fig. 4.7 Number of flipped edges in the predicted graphs, depending on the number of intervened variables; results of 50 random seeds with a confidence interval of 95%. Predictions are made by ALLIN, NOTEARS, PC-JCI GMM and PC.

For low-connectivity graphs with $\mathbb{E}[N] = 2$, the performance of the PC algorithm degrades even more with the size of $\mathcal{M}$, and already for only two intervened variables ($|\mathcal{M}| = 3$). Again, PC-JCI GMM does not suffer from this disadvantage, and even improves slightly for more intervened variables. This suggests that the method is able to exploit the presence of latent interventions to some degree, at least in the low-connectivity setting. In this setting, there is no statistically significant difference between ALLIN, NOTEARS and PC-JCI GMM. As pointed out in Section 4.2.1, this might be due to the limitations of ALLIN when the model assigns many datapoints to the interventional regime.

Theoretically, correctly detected interventions $\mathcal{I}$ would help to recover the true graph beyond the MEC, up to the $\mathcal{I}$-MEC. Specifically, interventions improve the prediction of edge orientations, i.e. they are expected to reduce the number of flips and undirected edges in the graph prediction. Figure 4.7 shows the number of flipped edges for each method, depending on the number of intervened variables, and Figure 4.8 shows the number of undirected edges. As expected, ALLIN and NOTEARS predict DAGs and, hence, no undirected edges. The flipped edges in the predictions of ALLIN are decreasing with the number of intervened variables,

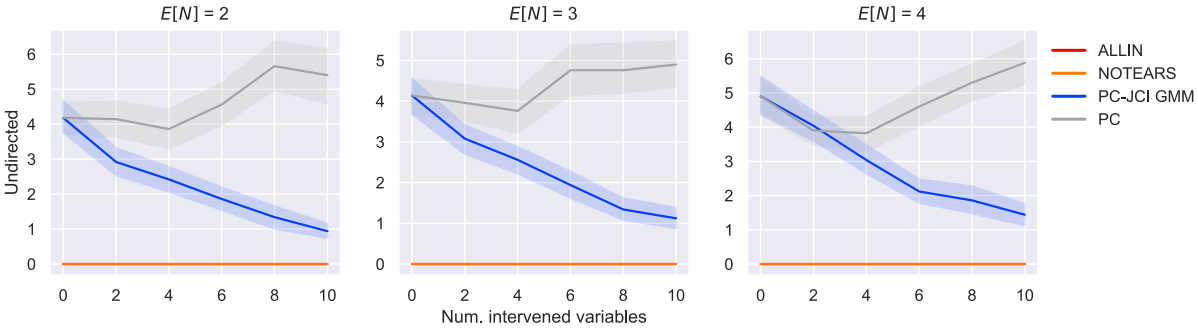


Fig. 4.8 Number of undirected edges in the predicted graphs, depending on the number of intervened variables; results of 50 random seeds with a confidence interval of 95%. Predictions are made by ALLIN, NOTEARS, PC-JCI GMM and PC. Since ALLIN and NOTEARS predict DAGs, the number of undirect edges is constant at 0.

but not consistently. This can partly be explained by the fact that the number of overall edge predictions is not consistent over the number of intervened variables for ALLIN. Generally, the number of flips is likely to be higher if more edges are predicted. When looking at the flips per predicted edge, the probability that a predicted edge is flipped decreases continuously with the number of intervened variables for graphs with $\mathbb{E}[N] \in \{3,4\}$ (see Appendix B). This indicates that ALLIN is indeed able to use interventional information to improve edge orientations.

Studying the number of flipped and undirected edges in the predictions of PC-JCI GMM reveals that this method fundamentally differs from ALLIN: for graphs with $\mathbb{E}[N] \in \{3,4\}$, the number of flipped edges increases more than the number of undirected edges decreases. This means that PC-JCI GMM actually worsens edge orientations when the number of intervened variables is higher. An exception to this are low connectivity graphs with $\mathbb{E}[N] = 2$, where PC-JCI GMM is able to orient undirected edges without an increasing number of flips for 6 or more intervened variables. Instead of improving edge orientations, the main advantage of PC-JCI GMM is that it protects the algorithm from detecting false positive edges (Figure 4.9), which pose the main challenge for the original PC algorithm on mixed observational and interventional datasets.

Hyperparameter dependency. ALLIN and NOTEARS share two hyperparameters: (1) the sparsity penalty λ and (2) the threshold ω. These hyperparameters are dependent on each other, i.e. the optimal value for ω changes depending on the value of λ and vice versa (Appendix A). For ALLIN, the optimal values for λ and ω are expected to differ from the optimal values for NOTEARS, due to the assignment of datapoints to the interventional regimes. Considering that these assignments change with every iteration of ALLIN, the dependency of the algorithm's performance on the hyperparameters becomes even more complex.

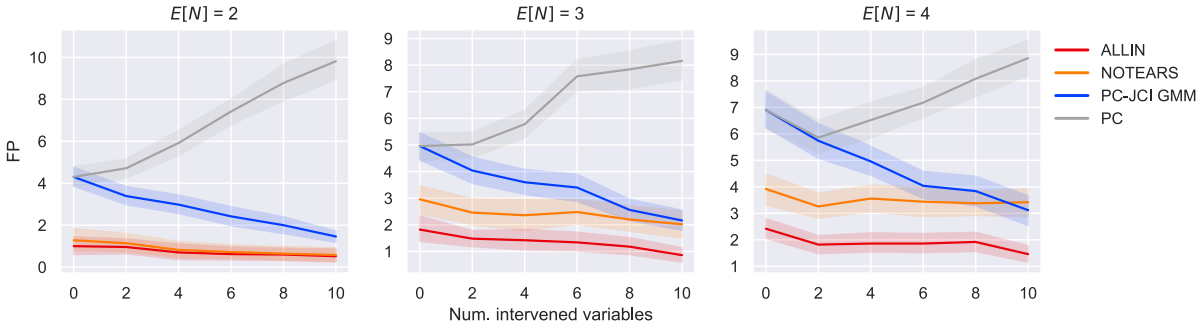


Fig. 4.9 Number of false positive edges in the predicted graphs, depending on the number of intervened variables; results of 50 random seeds with a confidence interval of 95%. Predictions are made by ALLIN, NOTEARS, PC-JCI GMM and PC.

Table 4.4 Optimal hyperparameters of ALLIN and NOTEARS on graphs with $\mathbb{E}[N] = 4$.

Method	**Num. intervened variables**	λ	ω
ALLIN	0	0.342	0.003
	10	0.188	0.004
NOTEARS	0	0.403	0.123
	10	0.573	0.046

Table 4.4 shows the result of tuning λ and ω for ALLIN and NOTEARS on 30 graphs with $\mathbb{E}[N] = 4$ and 0 or 10 intervened variables. As expected, the optimal value of λ is smaller for ALLIN than for NOTEARS. Additionally, the optimal value of λ for ALLIN depends on the number of intervened variables. For 10 intervened variables, the optimal value for λ is smaller than for purely observational data. A possible reason for this might be that ALLIN assigns more datapoints to the interventional regimes when more interventions are present in the data, which in turn weakens the signal for existing edges and therefore benefits from a smaller resistance through the sparsity penalty.

Due to the complexity of the hyperparameter dependencies, it is important to rule out that the improvement of ALLIN over NOTEARS is merely the result of a lucky choice of hyperparameters. Figure 4.10 shows the performance of ALLIN and NOTEARS with tuned hyperparameters (Table 4.4) on graphs with $\mathbb{E}[N] = 4$, depending on the number of intervened variables. For 2-8 intervened variables, the SHD is linearly interpolated. This comparison between the tuned versions of ALLIN and NOTEARS shows that the advantage of ALLIN over NOTEARS per-

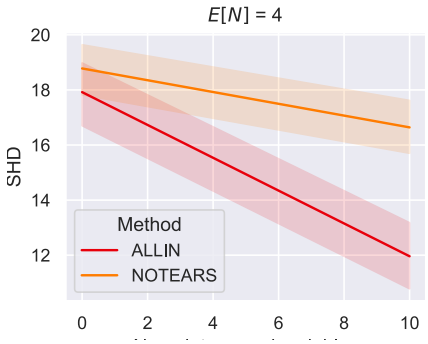


Fig. 4.10 SHD of ALLIN and NOTEARS with tuned hyperparameters.

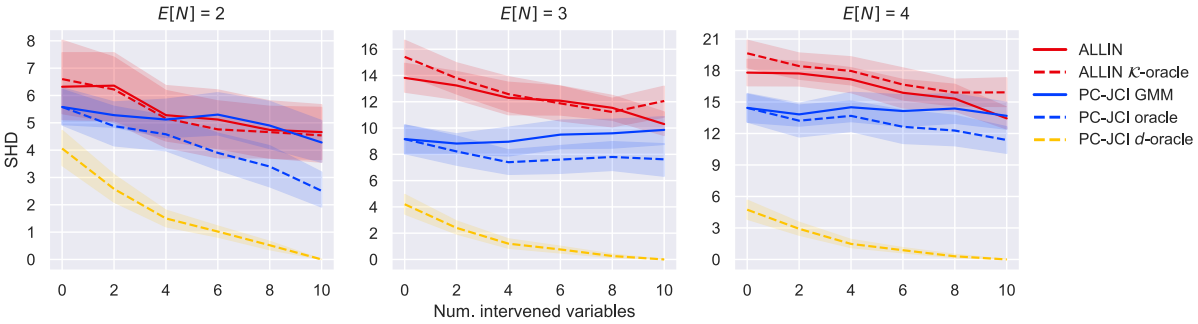


Fig. 4.11 SHD between the true graphs and the predicted graphs, depending on the number of intervened variables; results of 50 random seeds with a confidence interval of 95%. Predictions are made by ALLIN, PC-JCI GMM and their corresponding oracle versions with known interventions and unknown targets.

sists and is not the result of unfair hyperparameter choices. On the contrary, ALLIN's differential advantage seems to increase even further through tuning.

Known interventions, unknown targets. ALLIN and PC-JCI can both be applied to the setting of known interventions with unknown intervention targets. In this setting, the data is already separated into clusters according to the generating SCMs, but it is unknown which SCM characterizes each cluster. Thus, the intervention targets of each cluster need to be inferred.

The application of PC-JCI with clustering to this setting is straightforward: similarly to PC-JCI GMM, the cluster memberships are used as values for context variables. Instead of using the clusters returned by the GMM, we now use the ground truth clusters (*PC-JCI oracle*). As a reference, we add an oracle of PC-JCI that is given the correct d-separations for predicting the graph, such that it does not have to infer these with CI tests. The PC-JCI d-oracle returns the theoretically identifiable graph in the setting of known interventions with unknown targets.

For the ALLIN $\mathcal{K}$-oracle, we use the clustering version of ALLIN (Section 3.3.2), where the memberships of the $K = R+1$ ground truth clusters $\boldsymbol{\mathcal{K}}^* = \{\mathcal{K}_1^*, ..., \mathcal{K}_K^*\}$ are given as inputs to the model that learns the assignment to different SCMs. To learn the assignments, ALLIN performs 5 iterations. Judging from our experiments with ALLIN GMM (Section 4.2.1), this number of iterations is sufficient for ALLIN with clustering to converge.

Figure 4.11 shows the SHD between the true graphs and the predictions of these oracles, as well as the non-oracle versions of ALLIN and PC-JCI with clustering. The ALLIN $\mathcal{K}$-oracle performs very similar to the normal version of ALLIN. Hence, ALLIN is not particularly suitable for the setting of known interventions with unknown targets, as it is not able to exploit the ground truth grouping of the data. In comparison, the PC-JCI oracle improves over PC-JCI GMM, making it more suitable for the setting. However, given that the quality of the clustering of the GMM is a lot lower than the ground truth clustering (Table 4.2), it is surprising that the

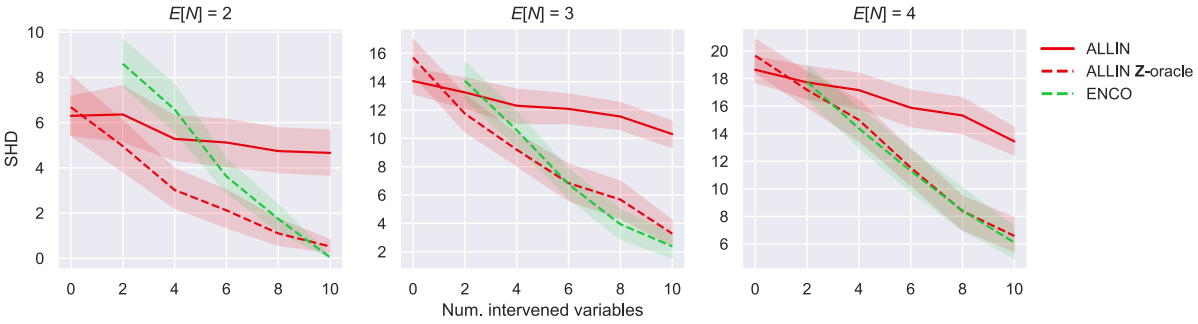


Fig. 4.12 SHD between the true graphs and the predicted graphs, depending on the number of intervened variables; results of 50 random seeds with a confidence interval of 95%. Predictions are made by ENCO, ALLIN and its corresponding oracle version with known interventions and known targets.

performance difference between PC-JCI GMM and the oracle is not larger. It follows that the ability of the PC-JCI oracle to infer intervention targets is still very limited, especially for the higher-connectivity settings with $\mathbb{E}[N] \in \{3,4\}$.

Generally, the PC algorithm is more prone to introducing errors in these higher-connectivity settings. This is not the case for the PC-JCI d-oracle, which shows a consistent performance on all connectivity settings. Since the PC algorithm tends to perform more CI tests when more edges are present in the graph, it is also more likely to make mistakes in the process. The results confirm that the limitations of CI testing pose the dominant bottleneck for the method rather than the clustering performance.

Known interventions, known targets. Figure 4.12 shows the performance of ALLIN with known assignments $\mathbf{Z}$ (*ALLIN* $\mathbf{Z}$*-oracle*). For $\mathbb{E}[N] \in \{3,4\}$, the oracle performs similarly to ENCO (Lippe et al., 2022), which is developed for the setting of known interventions with known targets on all variables. When interventions on all variables are present in the data, ENCO outperforms the ALLIN $\mathbf{Z}$-oracle on graphs of all connectivities. If not all variables are intervened upon, the ALLIN $\mathbf{Z}$-oracle outperforms ENCO for $\mathbb{E}[N] = 2$. However, the hyperparameters are close to optimal for ALLIN for $\mathbb{E}[N] = 2$, whereas they were not tuned for ENCO. Therefore, the comparison in this setting might not be fair. Nevertheless, ALLIN with known assignments performs competitively to ENCO, making it a suitable method not only for the setting of latent interventions, but also for the setting of known interventions and known targets. In general, the results show that ALLIN is able to use the information of interventional data for causal structure learning if datapoints are correctly assigned to their interventional regimes.

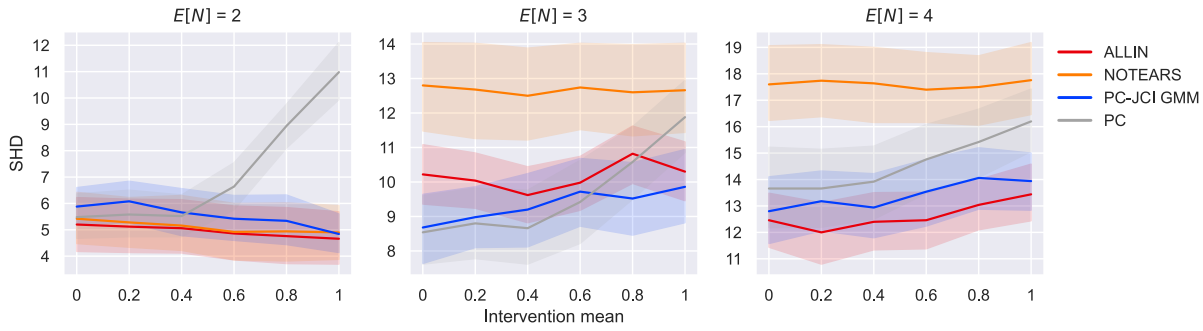


Fig. 4.13 SHD between the true graphs and the predicted graphs, depending on the interventional mean; results of 50 random seeds with a confidence interval of 95%. Predictions are made by ALLIN, NOTEARS, PC-JCI GMM and PC.

4.3.2 Interventional mean shift

In our setting, the mean of an interventional distributions is shifted compared to the observational distribution. To investigate the influence of this mean shift on the performance of our methods, we compare them on distributions with different intervention means, while keeping the mean of the observational distribution constant at 0. Since PC-JCI GMM relies on the separability of the distributions in the clustering space, we expect that the method benefits from a larger mean difference between SCMs. This differs from ALLIN insofar as ALLIN relies on the dropout of inputs for separating the data according to their generating SCMs. When the mean of the interventional data is not shifted, the model does not have to learn the biases $\mathbf{b}_{obs}$ and $\mathbf{b}_{int}$, such that it might even be easier for ALLIN to detect interventions without mean shift.

Figure 4.13 shows the performance of ALLIN, NOTEARS, PC-JCI GMM and the PC algorithm for different intervention means $\mu_{int} \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. Contrary to our expectations, PC-JCI GMM does not generally improve with larger intervention means. While a larger mean difference makes it easier for the GMM to separate the data according to the generating SCMs, this only results in a better performance for $\mathbb{E}[N] = 2$. For the higher connectivity settings, the performance of PC-JCI GMM decreases with a larger mean difference. This result is in line with our experiments from Section 4.3.1: only for $\mathbb{E}[N] = 2$, PC-JCI GMM improves with a higher number of intervened variables, which indicates that the method is only able to leverage latent interventions in this setting.

The performance of ALLIN is relatively consistent on all connectivity settings, within an SHD difference of up to 1. This confirms our expectation that ALLIN does not depend on the mean differences between SCMs for detecting interventions. On high connectivity-graphs with $\mathbb{E}[N] = 4$, the performance of ALLIN even increases slightly with lower mean differences, which could be related to the fact that the bias terms do not need to be learned.

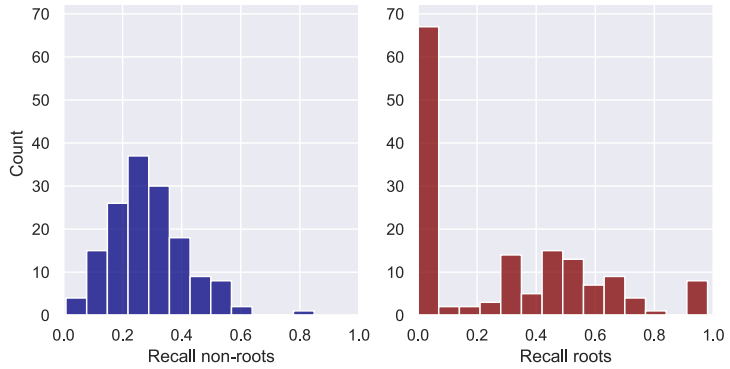


Fig. 4.14 Distribution of recall under ALLIN, i.e. the probability to detect interventions on non-root nodes (left) compared to root nodes (right). Statistics based on 150 runs from the main setting, including all three graph types with $\mathbb{E}[N] \in \{2,3,4\}$.

The method that stays most consistent over all intervention means is NOTEARS. This makes sense, as NOTEARS focuses on learning the observational SCM $\mathcal{M}^{(\varnothing)}$, which is unaffected by different intervention means. Meanwhile, the PC algorithm is the least consistent method: on all connectivity settings, its performance degrades when the intervention mean is larger than 0.4. This could be related to the false positive edges that pose a challenge for the PC algorithm, as spurious correlations can become stronger with larger mean differences.

Overall, the performance ranks of all methods except PC are consistent over different interventional means for each connectivity setting. For instance, on graphs with $\mathbb{E}[N] = 4$, ALLIN consistently outperforms PC-JCI GMM, which in turn consistently outperforms NOTEARS. This indicates that the advantages of PC-JCI GMM and ALLIN generalize to a wider range of settings.

4.3.3 Interventions on root nodes

Figure 4.14 shows the distribution of intervention recall under ALLIN in the main setting as described in Section 4.1.1. The value of the recall describes in how many experiments an interventional datapoint is assigned to the correct interventional regime, given that the variable is a root-node (right) or not a root-node (left) in the graph. For this, the assignment probabilities $p(\mathbf{x}_m)_i$ for each variable $X_i \in \mathcal{V}$ and each datapoint index $m \in \{1,...,M\}$ are rounded to hard assignments, i.e. 0 or 1, and the average recall is computed from these assignments for root nodes and non-root nodes separately. Then, the recall is again averaged over all experiments.

As expected, ALLIN has difficulties detecting interventions on root nodes: it does not detect any interventions on root nodes in about 45% of all experiments. To investigate, how severely this limitation could affect ALLIN, we test the performance of the ALLIN $\mathbf{Z}$-oracle

with known interventions and known targets when only non-root nodes are assigned correctly. The root nodes are assigned to the observational regime (as in 45% of our experiments).

Table 4.5 shows how the performance of this partial oracle compares to the ALLIN **Z**-oracle when all assignments are known. While the non-roots oracle does lose a significant amount of correctly identified edges, its performance is still superior to NOTEARS, PC and PC-JCI with clustering. Furthermore, it is uncertain if the performance of the partial ALLIN **Z**-oracle actually poses a lower bound of the SHD achievable by ALLIN, or if ALLIN has advantages beyond what can be captured with the oracle. However, the results indicate that ALLIN's limitation to detect interventions on root nodes indeed affects its performance.

Table 4.5 Performance of the ALLIN **Z**-oracle when all assignments are known (SHD, all), compared to knowing only assignments of non-root nodes (SHD, non-roots).

$\mathbb{E}[N]$	SHD (all)	SHD (non-roots)
2	0.52 ($\pm$ 1.07)	2.52 ($\pm$ 3.01)
3	3.28 ($\pm$ 3.29)	7.52 ($\pm$ 4.33)
4	6.58 ($\pm$ 4.38)	11.04 ($\pm$ 5.88)

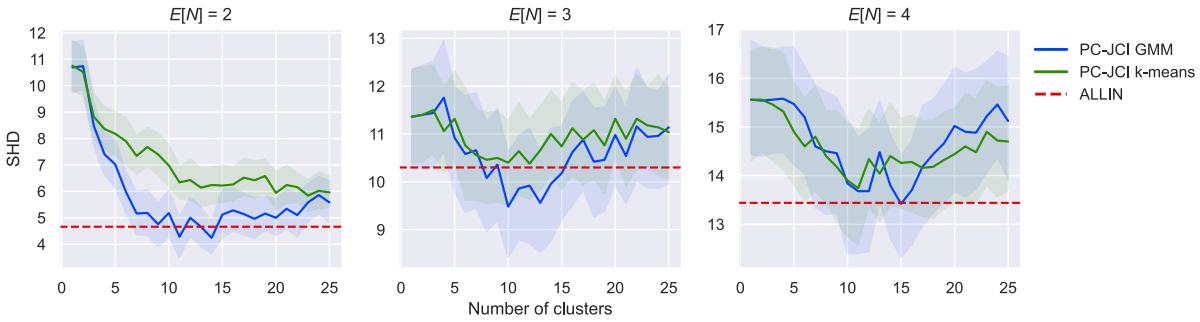
4.3.4 Model misspecification

In our experiments so far, our methods PC-JCI GMM and ALLIN perform similarly well, with PC-JCI GMM being superior in some settings. However, PC-JCI with clustering is subject to a number of limitations that are related to strong assumptions on the specification of the method, which are not required for ALLIN in the same way. In this section, we investigate these limitations on settings that result in a misspecification of PC-JCI with clustering.

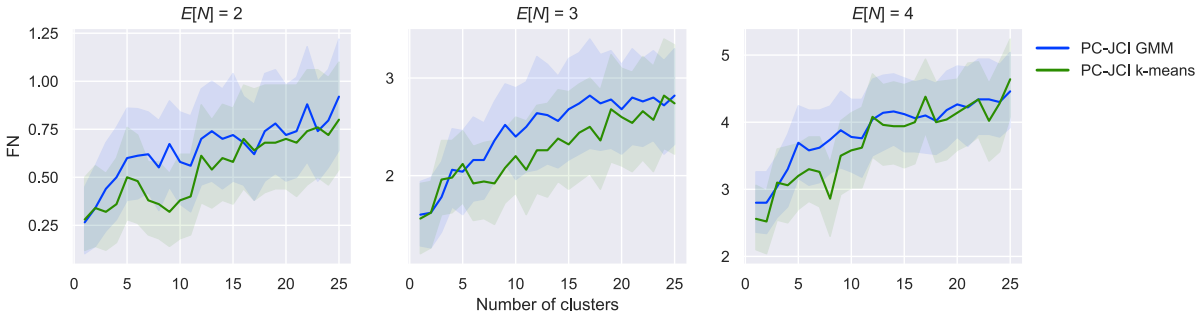
Number of clusters. As pointed out in Section 3.2.2, we expect the success of PC-JCI with clustering to depend on the choice of hyperparameters, specifically the number of clusters. Thus, we apply PC-JCI GMM to mixed data generated from 11 SCMs (as in the main setting), but misspecify the number of clusters in a range from 1 to 30. Additionally, we run these experiments with k-means as a clustering algorithm. When using the GMM, the true data-generating distributions are in line with the mixture model assumed by the clustering algorithm. However, this might not be the case in practice, therefore we include a setting in our experiments, where the distribution is misspecified.

Figure 4.15a shows the performance of PC-JCI GMM and PC-JCI k-means respectively, depending on the specified number of clusters. When using k-means clustering, the best performance is worse than for the GMM, such that PC-JCI k-means does not outperform ALLIN for any number of clusters. Furthermore, the performance of both PC-JCI with

clustering methods depends significantly on the number of clusters. Only for a small range, PC-JCI GMM and PC-JCI k-means perform well, which highlights the importance of a good choice of this hyperparameter, which might be a challenging in practice, as pointed out in Section 3.2.2.



(a) SHD.



(b) False Negatives.

Fig. 4.15 SHD (top) between the true graphs and the predicted graphs, depending on the number of clusters; results of 50 random seeds with a confidence interval of 95%. False negatives contributing to the SHD are shown on the bottom. Predictions are made by PC-JCI GMM and PC-JCI k-means. For reference, the mean SHD of ALLIN is added to the figure.

The contribution of false negative edges to the SHD is relatively small, as can be seen in Figure 4.15b. However, their number does increase with the number of clusters, which confirms that clustering can mask causal dependencies, as pointed out in Section 3.2.2. The largest contribution to the rising SHD for more clusters stems from an increasing number of flipped edges, i.e. incorrect causal directions. Responsible for the high SHD for a small number of clusters is a high number of false positive edges, which consistently decreases with the number of clusters for both the GMM and k-means. For a complete overview of the individual contributions to the SHD, see Appendix C. Since the setting with one cluster is almost identical with the normal PC algorithm, this illustrates how PC-JCI with clustering improves over PC: it compensates for the spurious correlations that are present in the mixture of observational and interventional distributions.

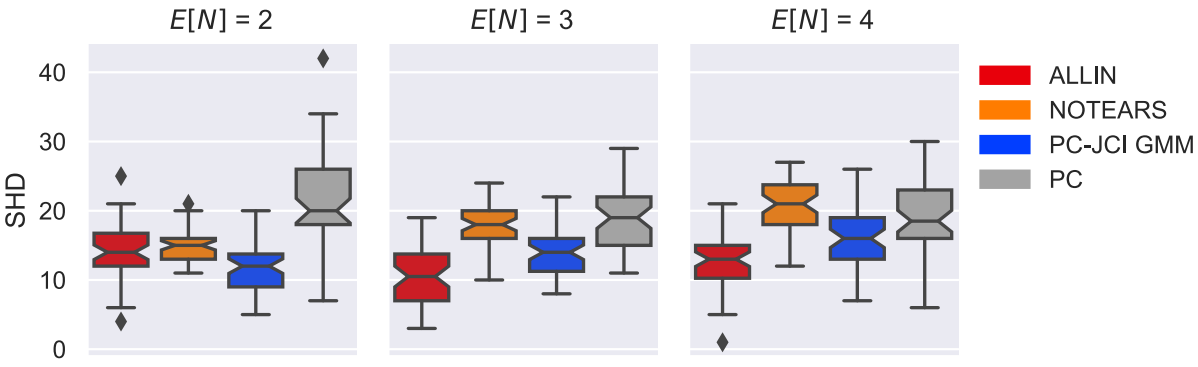


Fig. 4.16 SHD between the true graphs and the predicted graphs on multimodal data; results of 50 random seeds with a confidence interval of 95%. Predictions are made by ALLIN, NOTEARS, PC-JCI GMM and PC.

Multimodal data. If the data generated by one SCM is multimodal, a clustering algorithm might separate the modes within one SCM instead of separating the data generated from different SCMs. To investigate, how our method ALLIN, our baseline PC-JCI GMM and the existing methods, NOTEARS and PC, perform in a multimodal setting, we generate half of the data from an observational SCM as specified in the main setting, but shifted by -1, and the other half from the same observational SCM shifted by +1. This is equivalent to a single SCM $\mathcal{M}^{(\varnothing)} = (\mathcal{V},\mathcal{U},\mathcal{F})$ with a latent exogenous variable U_{mode} that is sampled from the following categorical distribution:

$$Pr(U_{mode} = +1) = 0.5 \tag{4.12}$$

$$Pr(U_{mode} = -1) = 0.5 \tag{4.13}$$

Depending on the value of U_{mode}, the endogenous variables take on the following values:

$$f_i : X_i = U_{mode} + U_i + \sum_{X \in Pa_i} X \tag{4.14}$$

For an interventional SCM $\mathcal{M}^{(I_k)}$, the intervention mean is identical with the main setting, such that the structural equations are:

$$f_i^{(I_k)} : X_i = \begin{cases} U_i + 1 & \text{if } i = k \\ U_{mode} + U_i + \sum_{X \in Pa_i} X & \text{otherwise} \end{cases} \tag{4.15}$$

Since U_{mode} appears in every structural equation, it is a latent confounder of the endogenous variables. Hence, the multimodality of the SCM introduces spurious correlations into the data,

which can affect all methods. However, PC-JCI GMM might be more affected as the data is more difficult for the clustering algorithm to separate correctly.

Figure 4.16 shows the SHD of the methods on multimodal data. On graphs with $\mathbb{E}[N] = 2$, PC-JCI GMM outperforms the other methods. On graphs with higher connectivities, ALLIN is superior. This confirms that PC-JCI GMM indeed suffers more in the multimodal setting, while ALLIN seems more robust to multimodality, reaching a performance on higher connectivity graphs that is close to the main setting with unimodal SCMs.

5 Discussion

In this thesis, we have presented two methods for causal discovery on mixed observational and interventional datasets with latent interventions: PC-JCI with clustering and ALLIN. PC-JCI with clustering divides the task of causal discovery on these mixtures of distributions into two separate subtasks, identifying interventions and discovering causal relations, such that, after detecting interventions, it can make use of existing causal discovery methods that operate on separated observational and interventional datasets with unknown targets. In contrast, ALLIN follows a joint approach, where it alternates between detecting interventions and learning the causal graph.

Sequential intervention detection and causal discovery. In experiments on datasets with perfect interventions, the presence of spurious correlations caused by the mean differences between distributions constitutes a significant challenge. For instance, we show empirically that a common causal discovery method, the PC algorithm (Spirtes et al., 1993), is particularly susceptible to interventional mean shift, with its performance suffering as the number of intervened variables and the magnitude of the mean shift increase. Our baseline method, PC-JCI with clustering, addresses this issue by resolving some of the spurious correlations through clustering, for example with a Gaussian mixture model (GMM). Our experiments show that the performance of PC-JCI with GMM clustering is consistent across different mean shift magnitudes and varying numbers of intervened variables present in the data. The robustness of PC-JCI with clustering towards the specific challenges of this setting allows the PC algorithm to perform effectively in the second part of the method.

One of the limitations of PC-JCI with clustering is that it not only resolves spurious correlations, but can also mask correlations that express directed edges in the true causal graph. This can occur when the number of cluster components is chosen too high, as each cluster adds a potential separating set to the graph prediction, each resulting in the possibility to predict a false negative edge. Additionally, we find no clear empirical evidence that PC-JCI with clustering uses the interventional information that is hidden in the datasets. Therefore, future improvements of this method might involve refraining from including cluster memberships

as context variables in the graph prediction and instead use the clustering as a basis for data-bootstrapping a causal discovery method of choice, such as the PC algorithm.

Joint intervention detection and causal discovery. In comparison to PC-JCI with clustering, our experiments indicate that ALLIN's joint approach of detecting interventions and discovering causal relations is more effective in exploiting the hidden interventional information: its performance increases when the number of intervened variables in the data is higher. In general, interventions provide information about edge directions. In line with this, an analysis of our experiments demonstrates that the largest contribution to ALLIN's improvement over more intervened variables is an improvement in edge orientations, showing that the algorithm makes use of the interventional information. A possible reason for the method's ability to exploit latent interventions is that it is based on assumptions on interventions that are not encoded in the approach of PC-JCI with clustering. It requires detected interventions to benefit from a dropout of input variables, and hence directly translates the definition of perfect interventions into its detection strategy. Compared to PC-JCI with clustering, this makes it more likely that a detected intervention is indeed a perfect intervention. However, it could also make the model blind to interventions of different types, which PC-JCI with clustering might be more capable of detecting.

An advantage of ALLIN over PC-JCI with clustering is that it implicitly performs clustering, but with a flexible number of clusters. ALLIN could potentially identify one cluster for each possible set of intervention variables. However, in contrast to clustering algorithms like k-means or GMM clustering, the model is not pushed to assign datapoints to each of these possible clusters. It only fills up a potential cluster with datapoints when the datapoints' characteristics align with the definition of perfect interventions. This enables the model to leave potential clusters empty, such that it is not susceptible to the limitation of PC-JCI with clustering of choosing an unsuitable number of clusters.

While ALLIN already improves causal discovery compared to existing methods in our experiments, it could still benefit from certain advancements. In particular, its performance depends on good hyperparameter choices of the incorporated NOTEARS algorithm. Instead of finding these hyperparameters through tuning, it would be better to automatically set these in a data-driven way. This is already suggested in Zheng et al. (2018) for advancing NOTEARS, but becomes even more important as the parameters would ideally be flexible across different iterations of ALLIN in order to adapt to the changes of each iteration. Furthermore, different strategies for detecting interventions can be explored. Currently, ALLIN uses a simple MLP to learn interventional assignments from the data. It is unclear how capable of detecting interventions this model is. The assignment learning step of ALLIN can potentially be improved

by choosing a different gradient-based assignment model. Another option is to replace the assignment model with a model-free approach that directly maximizes the data likelihood for given SCMs, similarly to the expectation step in the expectation-maximization algorithm. Lastly, the performance of ALLIN suffers when the model assigns too many datapoints to the interventional regimes. Therefore, it is important to modify ALLIN such that it avoids signal loss. One possible solution could be a rescaling of the gradients according to the number of effective observational datapoints.

5.1 Conclusion

Both our methods, PC-JCI with clustering and ALLIN, improve over the existing methods NOTEARS (Zheng et al., 2018), the PC algorithm (Spirtes et al., 1993) and DCDI-L (Faria et al., 2022) on synthetic linear Gaussian data generated from graphs with 10 variables. This indicates that PC-JCI with clustering and ALLIN are generally effective in detecting causal relationships from mixed datasets with latent interventions. However, the joint approach of ALLIN is superior when it comes to exploiting hidden interventional information. Our experiments with this method confirm that interventional data can be used to improve causal discovery, even if the interventions are latent. This result provides a positive outlook on the future of causal discovery from data collections that are commonly assumed to be observational. Through relaxing the observational assumption on such data collections, causal discovery can be improved beyond the limitations that exist for causal discovery from purely observational data, without requiring the expensive collection of controlled interventional data. Our method ALLIN provides a first step towards improvements in this direction.

References

Arthur, D. and Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA. Society for Industrial and Applied Mathematics.

Bellot, A. and van der Schaar, M. (2019). Conditional independence testing using generative adversarial networks. *Advances in Neural Information Processing Systems*, 32.

Bergsma, W. P. (2004). *Testing conditional independence for continuous random variables*. Citeseer.

Brouillard, P., Lachapelle, S., Lacoste, A., Lacoste-Julien, S., and Drouin, A. (2020). Differentiable causal discovery from interventional data. *Advances in Neural Information Processing Systems*, 33:21865–21877.

Chen, R. T. and DeStefano, F. (1998). Vaccine adverse events: Causal or coincidental? *The Lancet*, 351(9103):611–612.

Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554.

CMU-CLeaR (2021). causal-learn. https://causal-learn.readthedocs.io/.

Colombo, D., Maathuis, M. H., et al. (2014). Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15(1):3741–3782.

Faria, G. R. A., Martins, A., and Figueiredo, M. A. T. (2022). Differentiable causal discovery under latent interventions. In *First Conference on Causal Learning and Reasoning*.

Fisher, R. A. (1915). Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika*, 10(4):507–521.

Glymour, C., Zhang, K., and Spirtes, P. (2019). Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10.

Hauser, A. and Bühlmann, P. (2012). Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13(1):2409–2464.

Johnson, J. L., Lowenkamp, C. T., Van Benschotem, S. W., and Robinson, C. R. (2011). The construction and validation of the Federal Post Conviction Risk Assessment (PCRA). *Federal Probation*, 75:16–29.

Kaiser, M. and Sipos, M. (2021). Unsuitability of notears for causal graph discovery. *arXiv preprint arXiv:2104.05441*.

Kalisch, M. and Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8:613–636.

Kalisch, M., Mächler, M., Colombo, D., Maathuis, M. H., and Bühlmann, P. (2012). Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26.

Larson, J., Angwin, J., Kirchner, L., and Mattu, S. (2016). How we analyzed the compas recidivism algorithm.

Lippe, P., Cohen, T., and Gavves, E. (2022). Efficient neural causal discovery without acyclicity constraints. In *International Conference on Learning Representations*.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.

Mclachlan, G. and Basford, K. (1988). *Mixture Models: Inference and Applications to Clustering*, volume 38.

Meek, C. (1995). Causal inference and causal explanation with background knowledge. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, page 403–410.

Mooij, J. M., Magliacane, S., and Claassen, T. (2020). Joint causal inference from multiple contexts. *Journal of Machine Learning Research*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*.

Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

Pearl, J., Glymour, M., and Jewell, N. (2016). *Causal Inference in Statistics: A Primer*. Wiley.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rao, T. S. S. and Andrade, C. (2011). The MMR vaccine and autism: Sensation, refutation, retraction, and fraud. *Indian Journal of Psychiatry*, 53(2):95–96.

Reisach, A., Seiler, C., and Weichwald, S. (2021). Beware of the simulated dag! causal discovery benchmarks may be easy to game. *Advances in Neural Information Processing Systems*, 34:27772–27784.

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688.

Sampson, R. J., Morenoff, J. D., and Raudenbush, S. (2005). Social Anatomy of Racial and Ethnic Disparities in Violence. *American Journal of Public Health*, 95(2):224–232.

Shah, R. D. and Peters, J. (2020). The hardness of conditional independence testing and the generalised covariance measure. *The Annals of Statistics*, 48(3).

Spirtes, P., Glymour, C., Scheines, R., Kauffman, S., Aimale, V., and Wimberly, F. (2000). Constructing bayesian network models of gene expression networks from microarray data.

Spirtes, P., Glymour, C., Scheines, R., Spirtes, P., Glymour, C., and Scheines, R. (1993). Discovery algorithms for causally sufficient structures. *Causation, prediction, and search*, pages 103–162.

Splawa-Neyman, J., Dabrowska, D. M., and Speed, T. P. (1990). On the application of probability theory to agricultural experiments. essay on principles. section 9. *Statistical Science*, 5(4):465–472.

Strobl, E. V., Zhang, K., and Visweswaran, S. (2019). Approximate kernel-based conditional independence tests for fast non-parametric causal discovery. *Journal of Causal Inference*, 7(1).

UK Health Security Agency (2022). Confirmed cases of measles, mumps and rubella in England and Wales: 1996 to 2021.

Wiggins, B. (2020). Proxies. In *Calculating Race: Racial Discrimination in Risk Assessment*. Oxford University Press.

Zhang, K., Peters, J., Janzing, D., and Schölkopf, B. (2012). Kernel-based conditional independence test and application in causal discovery. *arXiv preprint arXiv:1202.3775*.

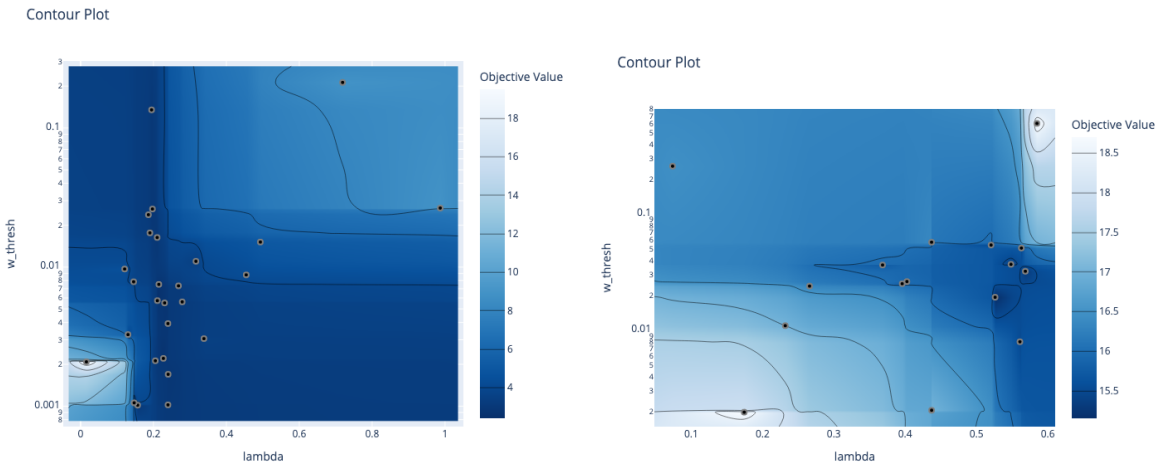
Zheng, X., Aragam, B., Ravikumar, P. K., and Xing, E. P. (2018). Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31.

Zheng, X., Dan, C., Aragam, B., Ravikumar, P., and Xing, E. (2020). Learning sparse nonparametric dags. In *International Conference on Artificial Intelligence and Statistics*, pages 3414–3425. PMLR.

A Hyperparameter tuning for NOTEARS

We conducted two hyperparameter sweeps for λ and ω with Optuna, on low connectivity graphs with $\mathbb{E}[N] = 2$ and on high connectivity graphs with $\mathbb{E}[N] = 4$. The sweeps involve runs on 30 random seeds each, which are different from the seeds used for our experiments and therefore correspond to different causal graphs.

Figure A.1 shows the contour plots estimated from the hyperparameter sweeps. Based on the estimated contour plots of the SHD, we choose the values $\lambda = 0.22$ and $\omega = 0.03$ that work for both settings, with a larger focus on graphs with $\mathbb{E}[N] = 2$, as the variance of the SHD seems larger in this setting. While the optimal hyperparameters are different for each setting, we refrain from using different values in our experiments. This corresponds to a more realistic approach, as the true causal graph's characteristics, such as sparseness, are unknown when applying a causal discovery algorithm to real-world data.



(a) $\mathbb{E}[N] = 2$ (b) $\mathbb{E}[N] = 4$

Fig. A.1 Estimated contour plots of the SHD, depending on λ and ω.

B Interventions on fewer variables

In this appendix, we provide more insights into the methods' behaviour in the ablation study with interventions on fewer variables. Figure B.1 shows the probability that a predicted edge is flipped in the true causal graph. Since the number of overall predicted edges varies for ALLIN, we normalize the number of Flips by the number of predictions, to show a clearer picture of an improvement in edge orientations. Figure B.1 shows that the probability of an edge being flipped steadily decreases with the number of intervened variables on graphs with $\mathbb{E}[N] \in \{3,4\}$, which indicates that ALLIN exploits interventional information to improve edge orientations. For graphs with $\mathbb{E}[N] = 2$, the orientations do not improve over NOTEARS, which we attribute to the limitations of ALLIN on low connectivity graphs.

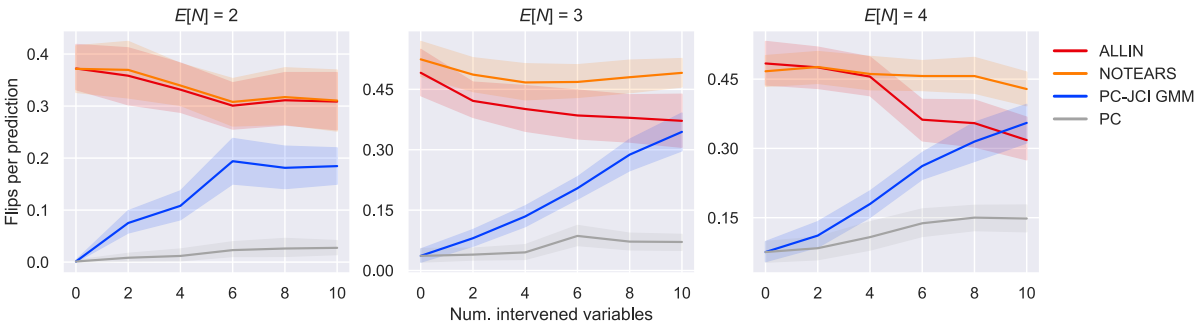


Fig. B.1 Number of flipped edges per predicted edge in the graph predictions, depending on the number of intervened variables; results of 50 random seeds with a confidence interval of 95%. Predictions are made by ALLIN, NOTEARS, PC-JCI GMM and PC.

Figure B.2 shows the number of overall edge predictions. The results on higher connectivity graphs demonstrate that ALLIN tends to predict fewer edges than NOTEARS, with a varying difference, depending on the number of intervened variables. These results do not only explain the noisyness of the absolute number of predicted Flips in Figure 4.7, but also indicate that ALLIN might indeed suffer from signal loss due to the assignment of datapoints to the interventional regime. While ALLIN seems to remove more incorrect than correct edges, it does lose a few correct edges on the way, as can be seen in the number of false negative edge

predictions in Figure B.3. Compared to the PC-based algorithms, ALLIN and NOTEARS both predict fewer edges in general, resulting in a higher number of false negatives.

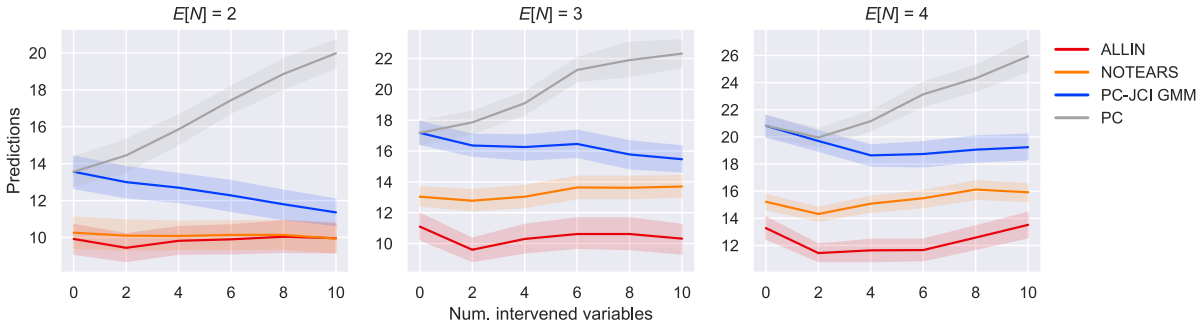


Fig. B.2 Number of edges in the predicted graphs, depending on the number of intervened variables; results of 50 random seeds with a confidence interval of 95%. Predictions are made by ALLIN, NOTEARS, PC-JCI GMM and PC.

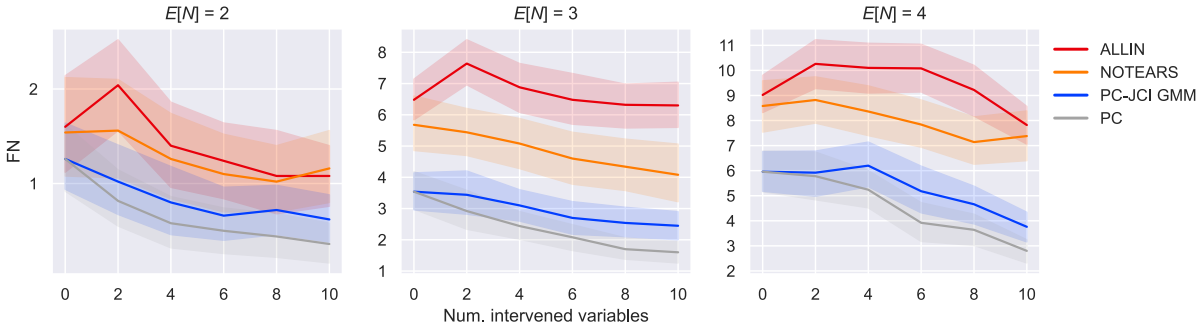


Fig. B.3 Number of false negative edges in the predicted graphs, depending on the number of intervened variables; results of 50 random seeds with a confidence interval of 95%. Predictions are made by ALLIN, NOTEARS, PC-JCI GMM and PC.

C Misspecified number of clusters

Figure C.1 shows the number of false positive edges for PC-JCI GMM and PC-JCI k-means, Figure C.2 the number of flips and Figure C.3 the number of undirected edges.

Fig. C.1 Number of false positive edges in the predicted graphs, depending on the number of clusters; results of 50 random seeds with a confidence interval of 95%. Predictions are made by PC-JCI GMM and PC-JCI k-means.

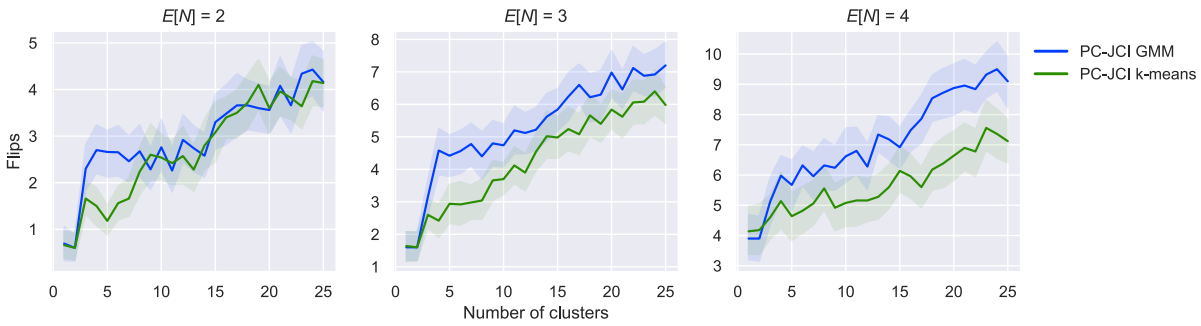


Fig. C.2 Number of flipped edges in the predicted graphs, depending on the number of intervened variables; results of 50 random seeds with a confidence interval of 95%. Predictions are made by PC-JCI GMM and PC-JCI k-means.

The individual contributions to the SHD in this misspecification setting confirm that PC-JCI with clustering resolves spurious edge correlations. When looking at the number of flipped edges, the edge orientations worsen with a higher number of clusters. The method orients more edges when the number of clusters is higher, but the decrease in undirected edges corresponds

almost perfectly with the increase in flipped edges. With every additional context variable in the extended causal graph, the number of potential v-structures increases, and therefore the number of edges that the PC algorithm can orient from the presence of unshielded colliders. However, the results indicate that this actually leads to a higher number of errors in the prediction of edge directions, instead of improving the orientations.

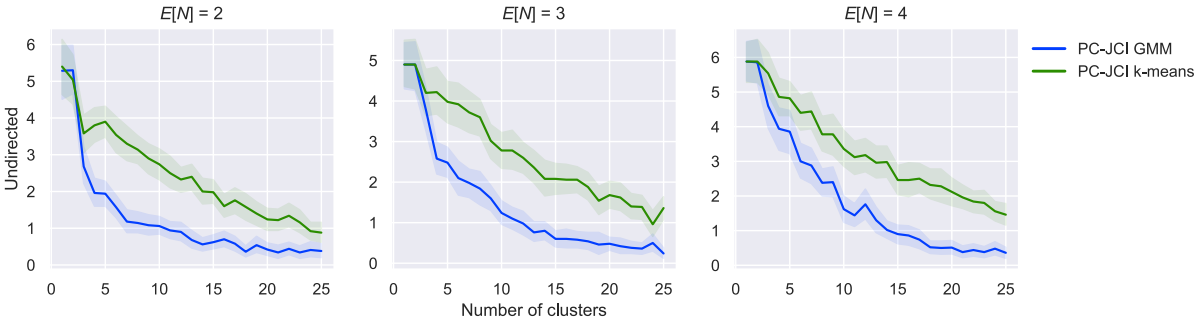


Fig. C.3 Number of undirected edges in the predicted graphs, depending on the number of intervened variables; results of 50 random seeds with a confidence interval of 95%. Predictions are made by PC-JCI GMM and PC-JCI k-means.