廣東工業大學
Guangdong University of Technology

A Report for Hybrid-based Causal Discovery with Machine Learning

Inferring Causation Under Generic Conditions: Nonlinearity and Confounders

Xuanzhi Chen

(Main) Supervisor: Wei Chen
Carnegie Mellon University & Guangdong University of Technology
Joint Training PhD on Machine Learning

School/Department of Computer Science
Bachelor in Computer Science and Technology

Undergraduate Research Projects on Causal Science (2021.09-2024.04)
Associated Preprinted Research Paper(s):
Chen, X., Chen, W.*, Cai, R., 2023. Non-linear Causal Discovery for Additive Noise Model with Multiple Latent Confounders. In Xuanzhi's Personal Website.* (Primary)
Associated Report as a Talk: https://www.youtube.com/@XuanzhiChen

Guangzhou, China

Declaration of Authorship

Has undersigned hereby declared that this work entitled “A Report for Hybrid-based Causal Discovery with Machine Learning”, is his own work and figures, illustrations , tables, equations, and code snippets in this report are original. It is also hereby declared that to the best of the knowledge, this work contains no material previously published or written by another person, except where the works of others have been explicitly acknowledged, quoted, and referenced. He understands that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

Since without adequate insights into AI-prompted writing, he also temporarily renounces the use of any AI generative content in this report, even if it may be beneficial to enhancing professionalism.

Guangzhou, China
Opensource Template Copyright 2024 @ School of Technology, Polytechnic University of Leiria

Xuanzhi Chen

Preface

It was an accident, but a happy one, for me to meet causation and complete relevant research work (X. Chen et al., 2023b; X. Chen et al., 2023a), where I spent roughly three years to study its moral, metaphor, and inspiration to machine intelligence. A substantial portion of my academic endeavors in undergraduate studies ended up with a preprinted paper entitled ”Non-linear Causal Discovery for Additive Noise Model with Multiple Latent Confounders”, as shown in the title of this report. During writing this report, I also always required myself: try to write only the ideas derived from your personal ”Eureka” moments in comprehending intuitions of causality. As a result, my personal taste affected the preference of topics. Judea Pearl (Pearl et al., 2018; Pearl, 2009) and Clark Glymour (Spirtes et al., 2000) initially shaped my view of the causality world, bringing insights into graphical causation unveiled by machine learning. Schölkopf, Bernhard (Peters et al., 2017) evoked my thinking about viewing the distribution nature entailed from structural causality as the hard problem in machine learning. Analyzing the causal rationale of these two types of methodologies for machine learning, my narration concerns their underlying causal assumptions implied by the causal model.

One shortcoming of this report is that I may be afraid of being able to comprehensively cover the entire research process through Chapter 4 ”*Literature Review*”, Chapter 6 ”*Programming (Code Samples)*”, and Chapter 7 ”*Results, Discussion, and Related Work*” in this report. This is in part because there is a two-year distance away from the most essential stage of the research work, with some scratch ideas and source code have been inevitably fuzzy according to my recollection. The other weakness lies in lacking time for crafting concise organization, since I decided relatively late in mid-November to write the report for my post-graduate application. The upcoming deadlines are issued.

Notwithstanding, I never write a research report for writing a research report. Admittedly, questions still swirling in my minds are, why do I posit myself seeming like writing for someone to read, given the fact that I am merely an undergraduate whose work is 99% to be considered as trivial? And, even for non-technical readers, there has been impressive and extraordinary learning materials such as “Introduction to causal inference” (Neal, 2020) in the field, so why do I make voice into the area in terms of the intuitions of causality? For the first question, I know my knowledge in causation will be inevitably dampen by time as I have graduated and left the field. Thus, at least I want to preserve them via text. I even felt fear for the second question when I was writing my personal essay (X. Chen, 2024) early in this year to popularize causality notions, but the lesson at worst I learned from that previous writing experiences is just that I know little about causation. I feel a modicum of comfort that in this report I kept making progress compared to the last time I did. So here I go.

Xuanzhi Chen
December 5, 2024

Abstract

Nearly all my undergraduate studies majoring in computer science consist in the research of causal discovery — developing model-based approaches capable of automating the discovery of cause-and-effect from observational raw data. When I was preparing for post-graduate study in November, 2024, I was recommended to supplement a written report regarding my past completed work on ”generic causal discovery” (e.g.causality theorem and algorithms applied into generic systems that possess nonlinearity and confounders). During my writing, I realize that the report may serve not only as a collection of my work, but an ultimate opportunity by which I am capable of contributing to the field for broader audiences interested in causality. I thereby hope that this report is able to complement extant work in two ways: **First**, it may reinforce the intuitions as to arguably two of the most essential philosophies/principles at the interface of causal discovery and Machine Learning (ML hereafter). Surrounding the key question ”how is data generated by Nature?”, I attempt to have this done by explaining the classic everyday-life examples whose prototype and causal metaphor lied behind used to confuse me when at that time I was a beginner in the field. **Second**, this report may provide a sample research workflow regarding a specific task of causal discovery, along with hands-on experimental details. Although I avoid to tout my work — a hybrid-based causal discovery framework — as an outstanding approach out of this report, it may be conductive to a better understanding if one tries to ”hybridize” different ML-provoked ideas under a united causality framework.

I make every effort to offer a systematic introduction into the topic, as well as my relevant research work, that is (at best) accessible to the readers with very basic statistical-ML knowledge and little insights into causal discovery. Therefore, different to my previous research papers, I adopt relatively informal language and flexible structures during writing, in particular the front part of the report. With that being said, I should state that it is overall meant to be written for (technical) readers who wish to make assessment on my undergraduate studies in the specialization of ML and causality.

Keywords: Causal Discovery, Machine Learning, Intuitions of Causality, Sample Research Workflow

Acknowledgements

Xuanzhi Chen would like to thank the DMIR (Data Mining and Information Retrieval) laboratory for offering him research opportunities, with special thanks to Ruichu Cai of the lab director and Dongning Liu of the dean in School of Computer.

In the DMIR lab, Postdoc Jie Qiao and graduate student Zhiyi Huang were willing to spend their time in personal discussions with Xuanzhi Chen about details in the paper; PhD student Zhengming Chen patiently helped point out initial mistakes in the paper; thanks to other prospective graduate students, such as Zeqin Yang, Zhengting Huang, Xiaokai Huang, and Yu Xiang, for their generosity of guidance when Xuanzhi chen commenced his internship in the lab

Specifically, Xuanzhi Chen owes a great debt to his advisor Wei Chen, for her encouragement when Xu- anzhi started studying causation two years ago — ”Do it, just keep your own interest and rhythm”

Xuanzhi Chen is also grateful to Dr. Judea Pearl and Dr. Clark Glymour, though he never meet them in person. Reading their works as a learner, one of the most exhilarating experiences is to realize that, merely by text, he could touch profound ideas in causality.

Finally, Xuanzhi Chen wants to thank all the potential readers who spend their time and patience in reading this report — He hope there are a modicum of causality intuitions useful for them.

Contents

Appendices

List of Figures

List of Tables

Glossary

Actual Cause . (*p. 5, 6*)
Average Causal Effect (ACE) . (*p. 4*)

Causal Additive Models (CAMs) . (*p. 21, 27*)
Causal Assumption . (*p. 4, 9, 10, 15, 16, 24, 46*)
Causal Asymmetry . (*p. 15, 25*)
Causal Beam . (*p. 5, 6*)
Causal Counterfactuals . (*p. 5, 6*)
Causal Discovery . (*p. 1–4, 6–13, 17, 18, 21, 24–28*)
Causal Graph . (*p. 3–9, 26, 29*)
Causal Intervention . (*p. 4*)
Causal Measurement Models . (*p. 6*)
Causal Model . (*p. 9, 10, 12, 13, 16, 17, 21, 27*)
Causal Preemption . (*p. 5*)
Causal Purity . (*p. 6*)
Causal Stability (or Perfect Mapness) . (*p. 9, 13*)
Completed Partially Directed Acyclic Graphs (CPDAGs) . (*p. 20*)
Conditional Independence Test (CIT) . (*p. 19, 25*)

D-Separation . (*p. 11, 12, 16, 17, 20, 32*)

Faithfulness Assumption . (*p. 9, 13, 17, 20*)
Full-Time Causal Graphs . (*p. 53*)
Functional Causal Models (FCMs) . (*p. 21*)

Independence of Cause and Mechanism (ICM) . (*p. 9, 15, 16, 18, 21*)
Independence Test . (*p. 26*)

Latent Confounder . (*p. 6, 19*)

Markov Assumption . (*p. 7, 9, 12, 13, 20*)
Markov Compatibility . (*p. 17*)
Markov Equivalent Classes (MECs) . (*p. 20*)
Maximal Ancestral Graphs (MAGs) . (*p. 20*)
Minimality Assumption . (*p. 9, 13, 17*)

This page is intentionally left blank for counting.

1

Introduction

1.1 Background

While the well-known Randomized Control Trials (RCTs) are widely accepted as the golden standard for Inferred Causation (Hariton et al., 2018), it is normally inaccessible due to the notoriously long durations or ethical concerns related to RCTs (e.g. it's unrealistic to experiment on healthy subjects to study whether years of smoking cause lung cancer) (Pearl et al., 2018). This is then what drives us to the question: Is it possible to automate the discovery of causation in the uncontrolled or observational raw data (as opposed to imposing human intervention in RCTs)? (Chapter 2)

1.2 Prerequisite

Over the last 40 years, there is excitement about progress at the intersection of Causal Discovery and machine learning (as well as causal thinking for AI). While the technical details thereof might be boring and daunting for readers unfamiliar with causation, the good news is, many conceptual ideas in Causal Discovery are actually closely tied to our common sense. (Chapter 3)

1.3 Related Work

Equipped with the insights into how machine learning techniques are capable of advancing Causal Discovery, two tracks of methodologies are then introduced in this report. My narration with respect to these classic approaches serves as a prelude to the work on hybrid-based Causal Discovery frameworks, which are developed by the DMIR (Data Mining and Information Retrieve) laboratory, arguably the first team in China studying causality learning. (Chapter 4)

1.4 Undergraduate Research Project(s)

My understanding of causality has primarily been shaped by Judea Pearl (Pearl et al., 2018; Pearl, 2009), Clark Glymour (Spirtes et al., 2000) and Schölkopf, Bernhard (Peters et al., 2017), with much of it went into my undergraduate work (X. Chen et al., 2023b; X. Chen et al., 2023a; X. Chen, 2024) that is (mainly) completed during my internship in the DMIR lab. (Chapter 5, Chapter 6, and Chapter 7)

1.5 Reading Guidance for Different Audiences

In an effort to increase readability of the report's content related to Causal Discovery, also to make my undergrad research work relatively accessible for broader audiences (if applicable), I thereby in this report adopted relatively **informal language and flexible structures**, avoiding abruptly inserting jargon without prior explanation. However, it's worth noting some of the content (nearly 60% of this report) are still meant to be written for technical readers — including experts, faculty, and/or employers — who wish to make assessment on my undergraduate research work.

Table 1.1 illustrates the associated reading guidance that allows the different types of audiences to navigate through this report. I also provided friendly "hooks" in each of primary sections by which readers can conveniently back to this table of reading guidance at any time they need.

Table 1.1: *Reading Guidance of This Report for Different Audiences.*

Chapter	Section	Technical Readers	Interested Readers
Chapter 2 (*Background*)	Section 2.1 - Section 2.3	O	✓
Chapter 2	Section 2.4	$\checkmark^*$	✓
Chapter 2	Section 2.5	✓	$\checkmark^*$
Chapter 3 (*Prerequisite*)	Section 3.1	$\checkmark^*$	✓
Chapter 3	Section 3.2	✓	$\checkmark^*$
Chapter 3	Section 3.3.1 - Section 3.3.3	✓	✓
Chapter 3	Section 3.3.4 - Section 3.3.5	✓	$\checkmark^*$
Chapter 3	Section 3.4.1 - Section 3.4.3	✓	✓
Chapter 3	Section 3.4.4 - Section 3.4.5	✓	$\checkmark^*$
Chapter 4 (*Literature Review*)	Section 4.1	$\checkmark^*$	O
Chapter 4	Section 4.2	✓	O
Chapter 5 (*Methodology*)	All Sections	✓	O
Chapter 6 (*Programming*)	All Sections	✓	O
Chapter 7 (*Results and Discussion*)	Section 7.1 - Section 7.2	✓	$\checkmark^*$
Chapter 7	Section 7.3	$\checkmark^*$	O
Chapter 7	Section 7.4	✓	O
Chapter 8 (*Personal Notes*)	–	O	O

Please kindly notice that there are three kinds of symbols $\{\checkmark, \checkmark^*, O\}$ in the table to help you judge the readability of the related sections:

- [✓]: The related section(s) is tailor-made for this type of audience.
- [$\checkmark^*$]: I am not ensure about the readability (and its reading value) of the related section(s) for this type of audience; it may be left for the readers themselves to determine:
  - either the technical readers may found it naive and decide to skip,
  - or the general interested readers may found it obscure and decide to skip.
- [O]: The related section(s) is optional for this type of audience.

Last but not least, although I try my best to cite relevant results, I don't meant to inform readers to dig into exactly which literature that contains the concept; I cite the result since this report may cover a diversity of readers, thus I try to be very careful about every idea I put forward in this report.

2

Causal Discovery: An induction Game Played Against Nature?

Trying to distinguish existing introductory materials related to the topic of Causal Discovery, one underlying purpose with which I organize this Chapter is to drive my readers to a big picture with respect to a wider implication of Causal Discovery. To put it alternatively, the odds are that finding out a Causal Graph — the objective of Causal Discovery in narrow senses — is relatively a small portion in comprehending causality. It may be helpful to further realize that the Causal Graph is fundamentally a kind of knowledge representation that highly generalizes a kind of relationship that is deterministic, irreversible, and thus stable. A common slogan in AI sufficiently summarizes my opinions made in this Chapter, which reads ”representation first, discovery second”.

Reminder: Several friendly hooks here helping direct you back to the previous Chapter 1, the next Chapter 3, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

2.1 Automating the Discovery of Causes

An exemplar around the broader discussion on observational raw data lies from where the Nature sits. Normally, one may assume that Nature by itself surreptitiously possesses a set of the established rules to manipulate variables in environment, out of which some rules, for example, may have been characterized as well-known physical equations such as the Laws of Thermodynamics and the Newton's Laws of Motion — But it's more than that. Widespread phenomena regarding predestined results complying with a certain rule, have long been in various philosophical debate on Causation.

Yet I, undoubtedly, have little knowledge about Philosophy. It's worth noticing that, however, akin to physicists who discover the physical law inductively from experimental data, the possibility of discovering causation merely from environmental raw data is thus a task that can be viewed as an induction game (Pearl, 2009) played against Nature — known as Causal Discovery.

2.2 Does Correlation Imply Causation?

One may argue that, nevertheless, statistical patterns automatically recognized by machine learning approaches do not necessarily imply causation. This common perception is fairly true though, a more rigorous statement is that ”Correlation Does Not (Logically) Imply Causation” (Pearl, 2009).

Thus, a more meaningful, and a more practical question is to ask, given what prerequisite does a machine learning technique being capable of prompting us to perceive a weaker relationship (Pearl, 2009) existing between the two? In terms of the mainstream progress, such prerequisites will be introduced in form of the several Causal Assumption in the next Chapter 3.

2.3 Conundrums in Causal Discovery

Although the Causal Assumption makes it theoretically possible to statistically discover structural causation, or the Causal Graph in terminology, one persisting conundrum consists in computational feasibility. By computational feasibility I mean the popular causality theory built upon the relationship between graphs and statistical distributions, which requires numerous algorithmic searches in both sides back and forth. As I will show in Chapter 4, it's natural for one to tease out the development of classic Causal Discovery algorithms in mid-1980s through their sharing target at attaining feasible computation. The other emerging track introduced in Chapter 4 is to resort to delicately fitting the functional composition between cause-and-effect. Thus another conceivable conundrum thereof lies in the delicate priori knowledge that is imposed on the complexion of causation.

It's worth noting that in above I merely introduce conundrums with respect to inferred theories in causality. When it comes to hands-on applications, impacts such as statistical errors in terms of the sensitivity and tolerance of Causal Discovery approaches over dataset are just as important, if not more, than these conceptual ideas (Don't ask me how I know that...).

2.4 Implications of Causal Graphs

Moving on, let's talk about the Causal Graph in this section, which is personally my favorite part in this report. While the (structural) applications of the Causal Graph seems uncommon in literature related to Causal Discovery, this report attempts to provide readers with thorough perspectives through which the roles of causal structures in the causation realms may become clearer. The following content in this Section is briefly rephrased and is referred from my online personal essay, *"A Primer on Causal Diagram Learning"* (X. Chen, 2024), that serves as the a kind of personal activity to popularize some valuable notions in the relevant topic.

Reminder: Several friendly hooks here helping direct you back to the next Section 2.5, the current Chapter 2, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

2.4.1 Intervention, Graph Surgery, and Causal Inquiry

One implication of the Causal Graph is to "translate" cause-and-effect inquiries from the perspective of causal diagrams — how to execute different "operations" on the causal diagram in light of different requirements for inquiry? For readers who don't mind a bit formula (Equation 2.1), by causal inquiries I mean the delicate difference (referred to as Average Causal Effect (ACE) for bi-variate cases) with respect to a situation with and without Causal Intervention:

$$ACE := P(y \mid do(X = x)) - P(y \mid do(X = x')). \tag{2.1}$$

In the context of probabilistic causation, the intervention expression is often formally denoted as $do(x)$ (reads ″do calculus″, meaning ″doing″ an action instead of ″observing″ an action). The value of the Causal Graph, namely a testable permission for answering the causal inquiry through graph operations, lies in that it foresightly implies the computational feasibility in terms of answering the (theoretical) causal inquiry merely through statistical estimation in the real world.

Metaphorically speaking, we can execute different types of ″surgery″ for the causal diagram to gain some intuitions of these inquires. Literally, two ″scalpels″ relative to the graph's nodes are called (in Judea Pearl's words) *″to hold it on constant″* and *″to tweak it on compulsion″*. ″Tweaking the node″ in a causal diagram refers to enforce an intervention, thus the other arrows pointing to the node waited to be tweaked will be deleted from causal diagrams — the only remained arrow suggests ″the tyranny of our human muscle″ to ″tweaking it″. Meanwhile, ″hold constant the node″ is similar to ″control″, but implicitly involves the Causal Counterfactuals information. This is because the most natural way to keep the (other) factors unchanged (while tweaking the main node) is to just render these factors to be *counterfactual* to the after-intervention situation.

Consequently, different kinds of the graph surgery mentioned above will result in different kinds of ″sub-graphs″. One insight into the sub-graph is to view it as the ″bed″ where specific criteria relative to the aforementioned ″testable permission in graphs″ can be applied. For instance, you may better understand my words if you have heard about the ″back-door criterion″ or ″back-door adjustment″ (Pearl et al., 2018) (incidentally, if there are more than one sub-graph, then multiple sub-graphs are tied to the well-known ″dynamic plan (James Robins, 1995)″ or ″sequential back-door adjustment (Pearl, 2009)″).

Relevant mathematical details can be found in my essay (X. Chen, 2024) (Section 3.2.1-3.2.2).

2.4.2 Counterfactuals, Causal Beam, and Attribution Analysis

The other implication of the Causal Graph its capability of attribution analysis and assisting one to discover a logically Actual Cause. Considering a causal link of ″mega-fire → climate change″, for instance, one singular circumstance can occur if the causal link is ″preempted″ by other causes such as human influence. By human influence I mean one may conceive a case where artificial drone can timely detect the mega-fire in Nature and prompt firemen to go and extinguish the fire. Thus, mega fire is a legitimate but not an Actual Cause, since the impact of a Causal Preemption is just about to invalidate its (the mega fire's) continuous causal effects to climate change, making the Actual Cause determination slightly elusive.

Interpreting by causal diagrams, the invalidation essentially amounts to expunge the arrows stemming from the cause that is preempted, which brings us to a revelation: In some of the singular scenarios, we can and should ″slim down″ a causal diagram by deleting some trivial arrows. Notice that this makes sense because some causal relations genuinely cannot exist when preemption accidentally occurs in that scenarios. Let us take a bit addition to our vocabulary to describe this behavior: ″slimming a Causal Graph down to a Causal Beam (Pearl, 2009). The Causal Beam, less rigorously, suggests that a causal graph is projected to a sub-graph relative to a singular scenario, serving as a switch in terms of the ″frame of reference″ of the surroundings. Therefore, the function of constructing the Causal Beam lies in helping us to clearly figure out the singular cause or, may be exactly the Actual Cause, by some technical transformation I do not show here.

Once again, for those readers who may not mind a bit formalism concept and have heard about the famous notion of ”necessary-and-sufficient causation” (Pearl et al., 2018), I can herein briefly introduce that the Probability of Actual Cause (PA) can be informally viewed as calculating the combination of both the Probability of Necessary Cause (PN) and the Probability of Sufficient Cause (PS) over the Causal Beam (let's denote it as $\mathcal{G}'$), instead of the Causal Graph (let's denote it as $\mathcal{G}$):

$$PA(X \to Y) = PN_{\mathcal{G}'}^{X \to Y} \cup PS_{\mathcal{G}'}^{X \to Y}, \ \ \{X, Y\} \subset \mathcal{G}. \tag{2.2}$$

Please notice that both PN and PS are causality concept relying on the Causal Counterfactuals information. This is because normally when we discuss the attribution analysis, we have the ”evidence” information beforehand. Both PN and PS calculus the information counterfactual to the current ”evidence” in probability context, which help us make a thorough assertion about the causes.

Relevant mathematical details can be found in my essay (X. Chen, 2024) (Section 3.1.1-3.1.2).

2.4.3 Causal Measurement Models

Most of the time, we are discussing Causal Discovery without the presence of unknown common factors (referred to as the Latent Confounder in terminology). The Latent Confounder is the unmeasurable variable in systems that affects more than one of other measurable variables, which poses significant challenge for Causal Discovery. The good news is, some insightful clues to causal inference are still available, and the idea behind them is about the ”trade-off” — the model constraints to simplify (a Causal Graph) usually brings the statistics strength to identify (the Causal Graph).

With a bit terminology, if one assumes that a Causal Graph represents the linear causal relationships with a more simple ”tree-like structure” than the regular graphs — also referred to as the Causal Measurement Models with Causal Purity in causal terminology — then such constrainted Causal Graph can bring us additional insights to identify its structure, even with the existing Latent Confounder. Such constraints have been mathematically characterized as the well-known Vanishing Tetrad Difference or the Tetrad Constraints, which can be found in the work *Causation, prediction, and search* (Spirtes et al., 2000) (Chapter 6, 10, and 11).

Similarly, relevant mathematical details can also be found in my essay (X. Chen, 2024) (Section 4.2).

2.5 A View Based on Bayesian Networks

I end up this Chapter with a classic perspective that discusses the relationship between a Bayesian Network (BN) and a Causal Network. The theory of Inferred Causation, consequently, is introduced in *Causality* (Pearl, 2009), with its author Judea Pearl being acclaimed as the inventor of BN.

Reminder: Several friendly hooks here helping direct you back to the previous Section 2.4, the next Chapter 3, the current Chapter 2, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

It is an insightful perspective to take the causal network, or roughly the Causal Graph, as a blueprint that connects different variables $\{x_1, x_2, ..., x_d\}$, and shows how the variable affects one another (e.g. $x_j \rightarrow x_i$ or vice versa) to represent how Nature may organize its rules to govern variables in environment (recall Section 2.1 early in this Chapter). Commonly, such perspective can be reached by adopt a BN (Stephenson, 2000). However, what's the difference between a regular BN and, more specifically, a causal BN? While the BN is commonly thought to be a statistical tool designed to reach inference under ”uncertainty” (Stephenson, 2000), it is anticipated at its best to be constructed under the moral of a ”certain” rule, namely the causal relationship (Pearl et al., 2018).

In light of this, one can carefully find that Conditional Independency (CI) occurred in everyday-life phenomena is in fact akin to a kind of by-product of the causal relationships stored in causal BN. To see that, let's consider a classic example (Pearl, 2009) in description of the relationships among whether rain falls (X_1), whether the pavement would get wet (X_2), and whether the pavement would be slippery (X_3). Since additionally introducing variable X_2 can quickly change our judgment on the relationships between X_1 and X_3 (That is, X_1 and X_3 are normally considered to be correlated at first, but then are judged to be uncorrelated after we have the information of X_2), we may deem such sensitivity to the ”granularity” of relationship as something hardwired to our brains, which implies the causal thinking useful for understanding Nature. We will see how this intrinsic intuition is mathematically characterized as the Markov Assumption when we move to Section 3.3 in the next Chapter.

To summary, Causal Discovery is equivalently meant to discover causal representation of a causal BN, which is presumed to characterize only causation (as compared to the general correlation). While a BN may occasionally be used as a directed cyclic graph to represent feedback cycles with industrial application, this report will focus the causal BN in form of Directed Acyclic Graphs (DAGs), also referred as to the Causal Graph, the Causal Diagram, or the Causal Structure.

3

Intuitions for Machine Learning Techniques on Causal Discovery

The Causal Graph introduced in the previous Chapter 2, as the objective of the Causal Discovery task, resembles a blue print that one can simply draw on a white sheet of paper, but what's the precise meaning behind the ”arrows” (→ or ← in the directed graph) it can tell us? Figure it out this point becomes important when it comes to talk about machine learning techniques, in part because these statistical tools may be expected to interpret their judgment on causal directions for good reason (e.g. $x_i \rightarrow x_j$ or vice versa). Hence, assuming a guideline at a high level is required.

Reminder: Several friendly hooks here helping direct you back to the previous Chapter 2, the next Chapter 4, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

3.1 The *Beuchet Chair* Optical Illusion

Take an example in description of the causal relationship between the objects' position (X) and their vision imaging (Y) perceived by our brains $(X \rightarrow Y)$. By the *Beuchet Chair* shown in Figure 3.1 we find that, given the right panel of the picture, our perception can go wrong if we posit such a single, carefully aligned, and very special vantage point to look at the chair (the two separated objects, in fact) — whereas most of time our vision perception holds well.

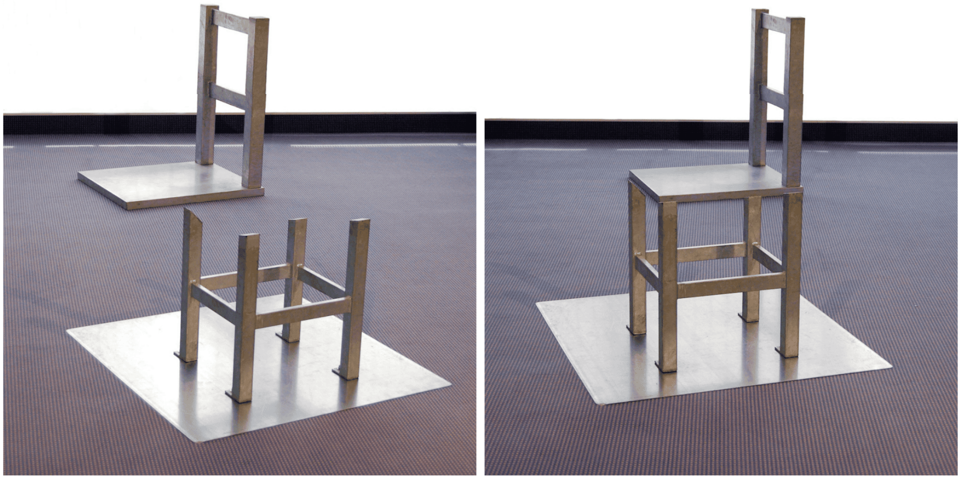


Figure 3.1: *Beuchet chair, made up of two separate objects (left panel) that appear as a chair (right panel) when viewed from a single, carefully aligned, and very special vantage point (Image courtesy of Markus Elsholz, reprinted from "Causality for machine learning" (Schölkopf, 2022)).*

Intuitions about most of the prerequisite for Causal Discovery algorithms is to either rule out such abnormal conditions (e.g. akin to the special vantage point) or reinforce normal mechanism (e.g. akin to the independent function of vision imaging, regardless the objects' position). Thus, moral of the *Beuchet Chair* can involves detailed discussion on the mainstream Causal Assumption, such as the Markov Assumption, the Minimality Assumption, the Faithfulness Assumption (Spirtes et al., 2000) (*Causation, prediction, and search, Section 2.3, 3.4*); the Causal Stability (or Perfect Mapness) Assumption (Pearl, 2009)(*Causality, Section 2.3, 2.4*); and the Assumption of Independence of Cause and Mechanism (ICM) (Peters et al., 2017)(*Elements of causal inference, Section 2.1, 4.1*).

It's unlikely to dive deep enough for so many aforementioned points in an introductory report; yet it may be a good start that leads to my opinions in this report shown in the following relative sections: "*Data Generation Process*" and "*Data Generation Mechanism*", which culminating with the basic insights relative to profound idea in Causal Discovery, namely to assume a kind of stable independency.

3.2 Causal Models

One way to lift up our viewpoint against learnable causal relationships, is to mathematically set up a parametric model that supplements a kind of "precise granularity" to the coarse Causal Graph $\mathcal{G}_X$. By granularity I mean the model specification regarding how each variable x_i (as an effect) is precisely affected by its parent(s) $pa(x_i)$ (as a cause) in the Directed Acyclic Graphs (DAGs).

Reminder: Several friendly hooks here helping direct you back to the previous Section 3.1, the next Section 3.3, the current Chapter 3, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

3.2.1 Mathematical Formalization

To this end, imagine a set of compatible parameters (denoted as $\Theta_{\mathcal{G}_X}$) storing the arbitrary mapping of functional relationships that Nature imposes between x_i and $pa(x_i)$ in the Causal Graph $\mathcal{G}_X$, leading to a pair of M_X (referred to as the Causal Model (Pearl, 2009)) that

$$M_X \;= < \mathcal{G}_X, \Theta_{\mathcal{G}_X} > . \tag{3.1}$$

Specifically, one can view $\Theta_{\mathcal{G}_X}$ as irreversible mapping (thus denoted as ':=' instead of '=') that

$$\Theta_{\mathcal{G}_X} = \{\boldsymbol{f}(\boldsymbol{X}, \boldsymbol{\varepsilon}) \mid pa(x_i) := f_i(x_i, \varepsilon_i)\}, \tag{3.2}$$

where symbol $\boldsymbol{\varepsilon} = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_d\}$ indicate the arbitrary (yet mutually independent) disturbance that we assume Nature intend to impose and thus perturb the (originally stable) causal relationships between x_i and $pa(x_i)$. It doesn't necessarily to keep Equation 3.1 and 3.2 in mind, as I will (implicitly) provide associated instances bit by bit in the rest of this Chapter; let's now quickly turn to the intuition and the rationale behind the Causal Model.

3.2.2 Intuition and Rationale

Notice that if it is possible for humans to inductively learn a structure that represents data, it may simultaneously justify the presence of a hidden process and mechanism, by which Nature probably

”reason” these raw data in the very beginning. Let me make it more clear: We assume Nature inherently possesses the Causal Model (shown in Equation 3.1) — an omniscient perspective for Nature but ignorant for us — that allows it to generate the raw data.

The advantages of establishing the Causal Model are at least three-fold in this report:

- First and foremost, specifying the complexion of the ”arrow” gives us a sufficient space to discuss which form of the Causal Assumption should be satisfied as prerequisite.
- Secondly, the double-tuple Causal Model hints at the hybrid-based algorithm framework introduced in this report, which covers both a causal structure as well as its functional composition.
- Regarding the moral of encapsulating the uncertainty or ”disturbance” in the Causal Model,
  - it involves one of the most insightful opinion that spreads over the causation realms (e.g. Causal Discovery, Intervention, Counterfactuals) (Pearl, 2009)(*Causality, Section 7.1*);
  - it is consistent with the classic causal notion in description of a ”pesudo indeterministic system” (Spirtes et al., 2000) (*Causation, prediction, and search, Section 2.5*).

3.3 Data Generation Process

As part of the Causal Model (M_X) shown in Equation 3.1, Directed Acyclic Graphs (DAGs) (could be donated as $\mathcal{G}_X$ in M_X) are actually in itself a generic-yet-weak structural assumption on the data generation process, in part because by the DAGs one may tell how the data is generated in order.

Reminder: Several friendly hooks here helping direct you back to the previous Section 3.2, the next Section 3.4, the current Chapter 3, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

3.3.1 A Classic Example: Water Flows in a Pipe

To figuratively understand how different orders affect a processing system (Spirtes et al., 2000; X. Chen, 2024), consider a piece of water pipe system, where i, j, and k are three of its junctions and each one has a valve-switch available to be controlled. Hence, given the direction of water flowing as a ”chain” path $i \longrightarrow k \longrightarrow j$ or a ”fork” $i \longleftarrow k \longrightarrow j$ path, one can ”deactivate” the connection between the junctions i and j (or specifically the water flow here) by turning off the valve at the junction k.

In contrast, given the direction of water flows as an ”inverted-fork” path $i \longrightarrow k \longleftarrow j$, one may found that the connection between the junctions i and j could be ”activated”, if one suddenly turns off the the junction k (or its ”downstream” parts) that is (are) initially supposed to be turned on in order to let the water flow pass. This ”activation” is alarming and should not be permitted in reality, as two currents are colliding with each other at the junction k, which may cause the pipe to burst!.

3.3.2 Patterns of Conditional Dependence-and-Independence

From the example we find that, ”deactivating” the existing water flow creates a pattern of ”conditional independence”. Here, the ”condition” refers to purposely turn off the valve. Conversely, ”activating” the non-existing crash of water flow is akin to a result of ”conditional dependence”. Such phenomena cover broadly the every-life events, if we interpret the directionality that is embedded in causal relationships is operating just as the same one in the flow of water.

Let's envision the causal information carried smoothly by a sequence of events, where at the middle of the process some events serve as mediators. Similarly to cut off the flow of water, interrupting a crucial mediator destroys the conductivity from cause to effect, leading to a state of independency.

It's worth noticing that, however, intentionally manipulating a ”collider”, as we have seen in the water flow example, also ignites a kind of ”unfriendly” dependency in events. For instance, independently flip two coins $(X1, X2)$ for 100 times, record the results, and rang a bell $(Y = 1)$ only when the result are tail (e.g. $X1 = 1$ or $X2 = 1$). If after experiment one only looks at the records with a bell-ranging, it's undoubtedly (yet kind of silly) that the two coins are correlated! In other words, there is fundamentally no any causal rationale behind such conditional dependency.

3.3.3 Statistical Patterns, Directed Graphs, and Machine Learning

In fact, both conditional dependence and independence are the statistical terminology that characterize how two variables are connected or disconnected, given the third variable. The notion of Conditional Independency (CI) has been maintained as ”the heart in causal modeling” (Pearl, 2009). If it's not easy to directly discover the causal relationships, alternatively mining the patterns relative to conditional dependence-and-independence is for good reason. This is because, from the perspective of data generation process, one can view these statistical patterns as the ”by-product” of causation.

In terms of the data generation process entailed by Directed Acyclic Graphs (DAGs), let's consider a simple directed structure $x_i \rightarrow x_k \rightarrow x_j$, for instance. The data generation process can be viewed as: from the exogenous nodes x_i influenced by factors outside of the system, through the endogenous node x_k shaped by the system itself, all the way to the leaf nodes x_j that represent the ultimate data points the system processes. In fact, it is such type of causal interpretation that eventually results in the ubiquity of DAGs models in machine learning applications (Pearl, 2009).

Transferring the above moral into DAGs, to better understand its generation process under certain control, one may be able to draw an analogy between the ”activated connections” among nodes in the context of DAGs, and the ”conditional dependency” among random variables in the context of probability and statistics. By the same token, ”deactivated connections” in DAGs are closely linked to the ”Conditional Independency (CI)” in probability and statistics.

Mathematics for the relationships between graphs and probability are summarized as the machine learning technique referred to as Probabilistic Graphical Model (PGM) (Koller, 2009). I here only interpret its elementary insights through the lens of Causal Discovery in this report.

3.3.4 Structural Insights with D-Separation

Keeping this instance of one-way data generation process, as $x_i \rightarrow x_k \rightarrow x_j$ top-down through the DAGs, is helpful to grab the crucial idea named D-Separation (”D” refers to ”directed”) (Pearl, 2009) that the causality-provoked Probabilistic Graphical Model (PGM) (Koller, 2009) rests on. The D-Separation criterion can best be recognized if one attributes causal meaning to the arrows consisting of paths (a sequence of consecutive edges) in the DAGs (Pearl, 2009; Pearl et al., 2018).

DEFINITION (INFORMAL)

The D-Separation Criterion: Two disjoint sets of nodes X_i and X_j in the DAGs are said to be directedly separated given another disjoint sets X_k, if and only if the connection between X_i and X_j is

1. "deactivated" by including all the chain-fork path, where
   (a) $X_i \rightarrow X_k \rightarrow X_j$ (chain path, or X_k is a mediator),
   (b) $X_i \leftarrow X_k \rightarrow X_j$ (fork path, , or X_k is a confounder);
2. NOT "activated" by including all the (generic) inverted-fork path, where
   (a) $X_i \rightarrow X_k \leftarrow X_j$ (inverted-fork path, or collider X_k, or V-Structure),
   (b) $X_i \rightarrow ch(X_k) \leftarrow X_j$, and $ch(X_k)$ refers to the child or descendant of X_k.

This brings us to the most important place in this section. If one doesn't have something like the D-Separation criterion available to be checked or be informed, he or she may have no idea about how to use the conditional dependence-and-independence — patterns recognized by machine from data — to re-construct a graph that is assumed to represent the data.

Instead, according to the D-Separation criterion, the Conditional Independency (CI) is categorized as a state permitting separation between two sets of nodes that is formed in the DAGs (e.g. chain path, fork path), whereas the conditional dependency is classified as a "non-separated" state that implies a V-structure (inverted-fork path or collided path) in the DAGs.

3.3.5 Markov/Minimality Properties: Necessary and Sufficient Conditions

As a result of aforementioned discussion, statistical implications of the D-Separation criterion lead to the Markov Assumption (Pearl, 2009)(Spirtes et al., 2000). One may have heard about the famous saying from temporary circumstances that, "*The future is independent of the past, given the present moment*", where the context of time is absolutely one-way and directed. This requirement can be also well applied to the context of one-way data generation process embedded in the Directed Acyclic Graphs (DAGs), which machine learning techniques on Causal Discovery are anticipated to meet.

CAUSAL PREREQUISITE (INFORMAL)

Causal Markov Assumption: The D-Separation implications for $\mathcal{G}_X$ (DAGs) in M_X (Causal Model) compatibly lead to conditional independence that distribution $P(X)$ satisfies:

$$P_{\mathcal{G}_X}(X): \ (X_i \perp\!\!\!\perp X_j \mid X_k)_{\mathcal{G}_X} \Leftrightarrow (X_i \perp\!\!\!\perp X_j \mid X_k)_{P(X)}, \tag{3.3}$$

The Markovian compatible distribution is thus denoted as $P_{\mathcal{G}_X}(X)$. Notice that in causality, we always use the symbol "$\perp\!\!\!\perp$" to denote the state where a variable is "independent" of the other (and thus $\not\!\perp\!\!\!\perp$ for dependence).

Equation delineates the following compatible circumstance: if (in the context of graphs) a variable x_i is d-separated from its non-descendant x_j given the parent(s) $pa(x_i)$, then (in the context of probability) the variable x_i is independent of its non-descendant x_j given the parent(s) $pa(x_i)$. Readers may find it easier to read when respectively replacing variable x_i, non-descendant x_j, and parent (s) $pa(x_i)$, with the "future", the "past", and the "present moment" — if they will.

One may know that one of defining feature of the Bayesian Network (BN) lies in its capability to represent the joint distribution with a sparse network structure. From the perspective of the Markov Assumption, this is because storing the information of Conditional Independency (CI) helps remove unnecessary connections in the structure. More fundamentally, from the perspective of causation, it providing a "necessary" condition to test and remove the dependence that are not genuine causal relationships — for example, the dependence that is induced by a existing chain-path (e.g. there is a mediator) or fork-path (e.g. there is a confounder) (mentioned in Section 3.3.4).

Given this sense, in order to further ensure that the remaining statistical dependence can be thus consistently interpreted as the genuine causal relationship, a "sufficient" condition (X. Chen, 2024), referred to as the Minimality Assumption (Pearl, 2009) (Spirtes et al., 2000), is required.

CAUSAL PREREQUISITE (INFORMAL)

Causal Minimality Assumption: Minimizing the total amount of possible Conditional Independency (CI) until all of them are only implied by the Markov Assumption.

Generally, making it clear that — a combination of the Markov Assumption and the Minimality Assumption in Causal Discovery roughly operates in a way resembling a necessary and sufficient condition — is enough for the causal intuition I want to share in this Section.

Specifically, the Minimality Assumption can be further understood by three folds: (1) "Minimality" equally means to maximize the independence implied by the Markov Assumption, since additional independence raised exclusively from the Markov Assumption is deemed to be non-causal. (2) Minimality entails a sense of "simplicity", which means if two structures well represent the same data, the simple one is preferable. (3) From the perspective of the Causal Model, the Minimality Assumption is established at the level of graphs $\mathcal{G}_X$. A stricter version established at the level of parameters $\Theta_{\mathcal{G}_X}$ is commonly referred to as the Faithfulness Assumption or Causal Stability (or Perfect Mapness).

3.4 Data Generation Mechanism

We now turn to the other part of the Causal Model (M_X) mentioned in Equation 3.1, where the composition of causal parameters $\Theta_{\mathcal{G}_X}$ (e.g. functional relationships and unmeasurable disturbance) actually has possessed an inherent assumption on the mechanism by which the data is generated.

Reminder: Several friendly hooks here helping direct you back to the previous Section 3.3, the next Section 3.5, the current Chapter 3, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

3.4.1 An Classic Example: Temperature and Altitude

Let me start explaining my opinion by borrowing another simple example of Causal Discovery (Peters et al., 2017) that describes the relationship between temperature (T) and altitude (A), where we all agree that a change (decrease) in temperature results from an increase in altitude:

$$\Delta T \leftarrow A. \tag{3.4}$$

Considering the "precise granularity" of this model, or its generation mechanism, let me explicitly describe their relationships in a physical equation:

$$\Delta T = -L \cdot A, \tag{3.5}$$

where L refers to the Lapse Rate (a parameter indicating the average rate of temperature decrease with altitude), serving as the major function that converts the influence from A and conveys it to T.

As I mentioned in the previous Section 3.3 "*Data Generation Process*", one can obviously point out that Equation 3.4 itself almost depicts a simplest bi-variable Directed Acyclic Graphs (DAGs), denoted as $\mathcal{G}_{\{T,A\}}$, in the context of probability. That is, one can estimates the joint distribution $P(A, T)$ from data samples collected in the different cities in different countries.

3.4.2 Differentiating Natural Causation from a Physical Equation

When it comes to the data collection in real environment, however, it's worth noticing that considering other factors responsible to the change in T — though minor but still conceivable — will make the data generation process more "natural" (as opposed to "ideal" or "without being erroneous"). Such factors might include atmospheric pressure (P) and humidity (H) processed by an unmeasurable physical mechanism $\sigma(\cdot)$, leading us to a more tolerant and nuanced transformation that

$$\Delta T = -L \cdot A + \sigma(P, H). \tag{3.6}$$

Distinguishing Equation 3.5 from 3.6 is important since the later characterizes something unexpected but inevitable when one samples data from the real world, which thus differentiates data generation followed by causal relationships from the one by purely physical description.

3.4.3 Inspiration from Physically Independent Mechanism in the Real World

Since we are talking about how to learn causal relationships in this report, our focus consists in the insights that we can draw from the data generation processed through physical mechanism.

Aside from the constant L, in Equation 3.6 one can tell that several trivial factors during a physical generation process, such as P and H, are always in place and lead to an influence on T, independent of A, or the city from which one choose to sample the data and form the distribution.

For example, during a process where one designs control-experiment in different regions to measure how temperature would change given a change in altitude, the mechanism ($\sigma(\cdot)$) or the physical way by which the ubiquitous atmospheric factors participate in the process, is basically the same — irrelevant to whether one choose to record T and A in Los Angeles or in Beijing (uh...as long as he or she conducts the experiment on Earth).

DEFINITION (INFORMAL)

Physically Independent Mechanism: A causal generative system akin to the system consisting of input-output modules (without feedback), where each module satisfies:

1. input ⟶ mechanism ⟶ output
2. mechanism = causal mechanism (major) ∪ physical mechanism (minor)
3. changing an input makes NO impact on mechanism, in particular the physical one

Imagine an input-output relationship is conductive to further comprehend the well-known Causal Assumption, Independence of Cause and Mechanism (ICM), made on data generation mechanism.

3.4.4 Independence of Cause and Mechanism: Intuition of Causal Asymmetry

The assumption of Independence of Cause and Mechanism (ICM) is a bi-variable version of the Physically Independent Mechanism, simplified with cause-and-effect relationships.

CAUSAL PREREQUISITE (INFORMAL)

Independence of Cause and Mechanism (ICM): Given two of the distribution associated with the cause-and-effect relationship (denoted as $P(C)$ and $P(E)$), $P(C)$ is independent of the conditional distribution of variable E given its cause C:

$$P(C) \perp\!\!\!\perp P(E \mid C). \tag{3.7}$$

Here the conditional distribution $P(E \mid C)$ can be interpreted as the minor mechanism, where the majority of causal information has been moved (conditioned) from the effect.

Notice that in causality, we always use the symbol "$\perp\!\!\!\perp$" to denote the state where a variable is "independent" of the other (and thus $\not\perp\!\!\!\perp$ for dependence).

A Thought Experiment on the Reverse Situation

Assuming the existence of Independence of Cause and Mechanism (ICM) entails an uniquely insightful property featured as Causal Asymmetry. To illustrate this, one just need to do a thought-experiment relative to the temperature-altitude example. Hypothesizing a reverse structure $T \rightarrow A$ this time, suppose one is then asked to picture a situation (Peters et al., 2017), where the altitude of a city dropped after a huge heating system was built around the city to raise the temperature. Not surprisingly, people would have a hard time thinking of such a reverse (and weird) situation.

At least the moral of this thought-experiment is straightforward: In order to distinguish between a result and a cause, one should comprehend that intervening the result can never affect the cause — only the opposite is absolutely true.

Causal Intervention

What the assumption of Independence of Cause and Mechanism (ICM) accounts for the aforementioned phenomenon is that, our cognition tend to be more adaptive to envision a consequence over a causal direction (Pearl et al., 2018; X. Chen, 2024) instead of an anti-causal one.

When we can quickly envision how the temperature in response to a rise of altitude without climbing the to mountain top in person, chances are, we have mentally (and often unconsciously) completed an intervention during this process. The independency of the mechanism implied by a causal direction retains the influence of those uncertain physical factors when a new change happen, thus enabling us to safely change the input (cause), and effortlessly predict the output (effect).

In contrast, when we have a hard time thinking over an anti-causal direction, it may imply that such mechanism is volatile, obscured, or even doesn't exist with a physically deterministic form. In Nature, temperature's change along with the seasonal switch can barely have an impact on an area's altitude, unless it's coinstantaneous to extreme events, such as the continental drift or the rise of sea level. Nevertheless, once such rare likelihood occurs, it showcases by no means the independency between its cause and (its associated) physical mechanism.

3.4.5 Explanation for Causal Asymmetry with Machine Learning

As a result, an interested point taken from the machine learning perspective is that, unlike the Independence of Cause and Mechanism (ICM) relative to the bi-variable relationship, one can barely talk about the machine learning techniques such as Probabilistic Graphical Model (PGM) without the presence of a "third-party". The D-Separation criterion I mentioned in the previous Section 3.3.4, for instance, delineates how the two sets of variables are counted as "blocked", given another disjointed variable set existing in the Directed Acyclic Graphs (DAGs). According to the Causal Assumption in Equation 3.7, the assumption of Independence of Cause and Mechanism (ICM) could be technically tested by regression-based machine learning techniques:

$$P(A) \perp\!\!\!\perp P(T - \hat{L} \cdot A), \tag{3.8}$$

if one applies Linear Regression (LR) over the recorded data to obtain an estimation of $\hat{L}$ concerning the lapse rate, and calculates $P(T - \hat{L} \cdot A)$ to approximate the conditional distribution $P(T \mid A)$ (according to Equation 3.7) that represents the effects of the independent mechanism $\sigma(P, H)$. At the same time, the Linear Regression (LR) can certainly be applied on a reverse direction, but it may probably left the machine with a "hard time" to fit the data, given the intuitive asymmetry we mentioned.

One may argue that it's straightforward to obtain the reversed relationship (Equation 3.6) as $A = \frac{-\Delta T + \sigma(P,H)}{L}$, by swapping the variables T and A on both side of the equality sign. Notice that, however, in our context I implicitly assumed the linear functional relationship (by a constant Lapse Rate) between T and A, and have not yet made any explicit form of $\sigma(P, H)$ relative to the physical mechanism. These considerations are, in fact, crucial, when it comes to the practical feasibility of using regression-based machine learning techniques to discover causal relationships, since the causal mechanism $\Theta_{\mathcal{G}_X}$ encompassed by the generic Causal Model (I mentioned in Section 3.2) may be a (massive) set of parameters needed to be determined (in a more complex multivariate circumstance). Details about this point are left for some interested technical readers in Chapter 4, "*Literature Review*".

3.5 A Holistic Review Based on Bayesian Networks

This section is meant to additionally clarify some relevant notions in causality, in particular echoing to the Bayesian Network (BN) mentioned in Chapter 2, "*A View Based on Bayesian Networks*".

Reminder: Several friendly hooks here helping direct you back to the previous Section 3.4, the next Chapter 4, the current Chapter 3, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

3.5.1 Data Generation Process: Markov Compatibility

One defining feature of the BN, in terms of its graphical structure, is to facilitate economical representation (Koller, 2009) of the joint distribution $P(\boldsymbol{X})$ over variables, as shown in Equation 3.9.

$$P_{\mathcal{G}_X}(\boldsymbol{X}) = \prod_{x_i \in \boldsymbol{X}} P_{\mathcal{G}_X}(x_i \mid pa(x_i)), \tag{3.9}$$

where for any two of variables x_i and x_j, the distribution $P(x_i \mid pa(x_i))$ is independent of the one $P(x_j \mid pa(x_j))$, also concisely denoted as "$P(x_i \mid pa(x_i)) \perp\!\!\!\perp P(x_j \mid pa(x_j))$". The symbol $P_{\mathcal{G}_X}$ further indicates the Markov Compatibility that serves as a condition for a (causal) BN to capture a stochastic process capable of generating $P(\boldsymbol{X})$ (Pearl, 2009). Ascertaining this compatibility is important, because it enables us to model the relationships between graphs and probabilities

DEFINITION (INFORMAL)

Markov Compatibility: A joint distribution $P(\boldsymbol{X})$ is said to be compatible with a Directed Acyclic Graph $\mathcal{G}_X$, if and only if $P(\boldsymbol{X})$ implies the independency decomposition (shown in Equation 3.9) relative to $\mathcal{G}_X$, and thus can be denoted as $P_{\mathcal{G}_X}(\boldsymbol{X})$.

Moreover, such independency decomposition, or probabilistic factorization with in terminology, can be in fact alternatively, and thus conveniently, approached by listing a table of the Conditional Independency (CI) relationships that the Markov compatible distributions are anticipated to satisfy, according to the D-Separation criterion mentioned in Section 3.3.4. In short, the Markov Compatibility provides the rationale regarding the (structural) data generation process for a machine learning technique to identify; the D-Separation criterion helps the Causal Discovery algorithm realize the computational feasibility when mining those CI patterns.

Finally, even ensuring the the Markov Compatibility, there may still remain an unbounded number of the (causal) BN that can fit a given distribution. This makes it for good reason to impose a structural restriction such as the Minimality Assumption and/or a parametric restriction such as the Faithfulness Assumption, which were briefly mentioned in Section 3.3.5.

3.5.2 Data Generation Mechanism: Autonomy and Intervention

Considering the Causal Model as a whole, one can combine Equation 3.6 with (the prior structure $\mathcal{G}_{\{T,A\}}$ in) Equation 3.4 to obtain a simple bi-variable instance of the Causal Model (denoted as $M_{\{T,A\}}$) in relation to the example mentioned in Section 3.4.1, as shown in Equation 3.10 that

$$M_{\{T,A\}} \; = < \mathcal{G}_{\{T,A\}}, \Theta_{\mathcal{G}_{\{T,A\}}} >, \tag{3.10}$$

where $\Theta_{\mathcal{G}_{\{T,A\}}} = \{L, \sigma(P, H)\}$ involves compatible parameters entailed by Equation 3.6. From the instance we can now perceive that there is a property of the Independence of Cause and Mechanism (ICM) that serves as a principle implicitly assumed by the bi-variable structure $\mathcal{G}_{\{T,A\}}$.

Moreover, the insights into Physically Independent Mechanism in Nature and the physical world are not exclusively applied in Causal Discovery. In contrast, it closely involves other crucial concepts in the realms of causation. For example, assuming such independency grants a causal BN $\mathcal{G}_X$ (in Equation 3.1) the capability of making response to external changes with just economical modification over its network structure (Pearl, 2009). This is in part because, roughly speaking, the network connections that are physically glued by causation are more likely to be considered as robust, stable, and thus needless to (too much) training or adaptation. Technically, the implications of Physically Independent Mechanism coherent to the causal BN's properties are at least two aspects:

1. **Autonomy**: It facilitates deterministic relationships resembling from inputs to outputs, which distinguishes the causal BN from the regular BN. For example, unlike regular Bayesian techniques such as Brief Propagation (Koller, 2009) that blindly spreads over the entire network, the independency of mechanism entails local autonomy (Peters et al., 2017)(Pearl, 2009).
2. **Intervention**: Unimpacted mechanism enables a causal BN to estimate effects the Intervention (Peters et al., 2017) by truncated probabilistic factorization (with *do* calculus)(Pearl, 2009).

Diving into this point leads us to the high-level semantics in ares of causation such as (Structural) Intervention and (Structural) Counterfactuals, which goes beyond the scope (statistical causal discovery) in this report. Interested readers may refer to the relevant literature for more information.

4

Literature Review

Background narration and methodology delineation about the state-of-the-art approaches — classic Non-Temporal causal discovery algorithms in Section 4.1 — are relatively and partially summarized from one of my undergrad research work "A Survey on Causal Discovery with Incomplete Time-Series Data" (X. Chen et al., 2023a). Additionally, in order to unify the one of the causal discovery methodology developed initially by the DMIR (Data Mining and Information Retrieval) lab that I worked for, Section 4.2 further teases out the algorithms that rely on hybrid-based frameworks — the framework in which merits of the classic causal discovery algorithms in Section 4.1 are integrated.

While the Chapter in this report adopted relatively informal language and flexible structure, it's still written for technical readers — including experts, faculty, and/or employers — who wish to make assessment (regarding field knowledge and specialization) on my undergraduate research work.

Reminder: Several friendly hooks here helping direct you back to the previous Chapter 3, the next Chapter 5, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

4.1 State-of-the-art Approaches for Causal Discovery

4.1.1 The SGS, PC and FCI Algorithms

Background Narration

In the early 1980s, researchers like Glymour and Spirtes developed efficient causal discovery algorithms that utilize statistical patterns of (conditional) independence and incorporate the (structure) completeness through philosophical logic rules.

The methodology leveraging the Conditional Independence Test (CIT), known as the constraint-based approach, extends structural learning methods for the Bayesian Network (BN) within causal significance constraints. The machine learning intuitions as to this track of causal discovery algorithms can be found in Section 3.3 "*Data Generation Process*" in this report.

Methodology Delineation (In light of My Personal Critique)

Given the assumption of Causal Sufficiency (Spirtes et al., 2000) — causal discovery without the presence of latent variables (Latent Confounder) — fundamental approaches encompass the SGS (Spirtes-Glymour-Scheines) algorithm and the PC (Peter-Clark) algorithm (Spirtes et al., 2000).

Equation (4.1) represents the (notion/idea/strategy of the) CIT (in causal discovery): For any extant variable pair x_i, x_j, the algorithms will test their correlation (e.g. whether x_i, x_j are correlated or independent of each other) condition upon (every subset consisting of the) variables other than x_i, x_j within the observed variable set V.

$$CI\left(x_i,\ x_j \mid subset\left(\boldsymbol{x}^{V\setminus\{x_i,\ x_j\}}\right)\right). \tag{4.1}$$

Since it's possible to have multiple causal structures that can share identical Conditional Independency (CI) patterns (Spirtes et al., 2000) (aka. the Markov Equivalent Classes (MECs)), leading to partial orientation, namely some of the edges in the graph cannot be oriented to the arrows. This brings us to a notion named the Partially Directed Acyclic Graphs (PDAGs) that are relative to the Directed Acyclic Graphs (DAGs). So the goal (of the constraint-based approach) is to construct the MECs over the PDAGs, also known as the Completed Partially Directed Acyclic Graphs (CPDAGs).

Constraint-based methods rely on the Markov Assumption and the Faithfulness Assumption (with the assumptions' intuitions mentioned in Section 3.3.5 "*Markov/Minimality Properties: Necessary and Sufficient Conditions*", decomposing the learning process into two stages:

1. The skeleton learning stage relative to the Un-directed Graphs (or are referred to as the MRF, Markovian Random Field (Koller, 2009))
2. The orientation stage based on the "V-structure" (I've mentioned in Section 3.3.4 *Structural Insights with D-Separation* in this report) and the logic rules, relative to the DAGs

Commencing with a complete graph, execution of CIT eliminates redundant edges between pairwise variables, yielding a causal skeleton. The algorithms then orient the edge direction mainly in light of the V-structure provided by the condition (D-Separation) set, ultimately leading to CPDAGs.

In terms of systems that fail to satisfy the Causal Sufficiency assumption, the mainstream approach, represented by the FCI (Fast Causal Inference) algorithm (Spirtes et al., 2000), is adopted and proven to be theoretically correct, sound, and complete. The FCI algorithm is an extension of the PC algorithm, which introduces the concept of Maximal Ancestral Graphs (MAGs) and Possible D-separation Set (Possible-Dsep Sets) to aid in respectively representing the causal graphs and testing conditional independence in the presence of latent confounders. On one hand, similar to the PC algorithm, the FCI algorithm also employs the V-structure and the logical rule to determine causal direction over MAGs, leading to the search of Partially Ancestral Graphs (PAGs) that are analogous to PDAGs.

4.1.2 The CAM, LiNGAM, and ANM Algorithms

Due to the inherent challenges of Markov Equivalent Classes (MECs), structural uniqueness is still not guaranteed. Aside from imposing the Markov Assumption and the Faithfulness Assumption on the data generation process, the other track in mainstream causal discovery methodologies as mentioned in Section 3.4 typically resort to for a finely grained expression of causal relations.

Background Narration

Roughly starting in 2005, researchers including Hoyer and Shimizu developed a new scheme for inferred causation, which making delicate assumptions on the functional composition relative to a deterministic causal relationship. By functional composition I mean a well-known subclass relative to the Causal Model, referred to as the Causal Additive Models (CAMs) (e.g. literally indicating that the composition of functional influence that imposes on a variable is additive).

The core of methods (in such categories), is known as on the basis of Structural Causal Models (SCMs) or Functional Causal Models (FCMs) (in causal terminology), which interprets a causal system as a series of special equations — each of them explains the generation of variables as a result of their direct causes and independent noise terms, mapped through an irreversible causal function. Moreover, if researchers wish to explicitly discover causal relationships only from observational data while at the same time to ensure that the learned causal structure is unique, then additional assumptions must be added to the SCMs or FCMs. Similar to the constraint-based approach mentioned in the previous Section 4.1.1, the machine learning intuitions as to this track of Causal Discovery algorithms can be found in Section 3.4 "*Data Generation Mechanism*" in this report.

Methodology Delineation (In light of My Personal Critique)

One typical assumption imposed upon SCMs reads the Linear non-Gaussian Acyclic Model (LiNGAM) (Shimizu et al., 2006; Shimizu et al., 2011), which relies on the non-Gaussianity of noise. The basic idea of the LiNGAM is that the asymmetry (a kind of geometric constraint) inherent in the non-Gaussian noise mathematically allows for the causal identification that traditional Linear Gaussian Bayesian networks cannot achieve. In the LiNGAM, the SCM is represented as follows:

$$\mathbf{X} := B\mathbf{X} + \mathbf{N} \;\; \text{or} \;\; x_i := \sum_{i<j} \beta_{ij} x_j + \varepsilon_i. \tag{4.2}$$

where B and $\mathbf{N}$ (or β_{ij} and ε_i) represent the lower-triangular causal adjacency matrix (or causal strength coefficients from x_j to x_i thereof), namely the FCMs graphical structure in form of the matrix, and the non-Gaussian independent noise vector that together generate the observed variable vector $\mathbf{X}$ (or x_i). The mainstream methods for solving the LiNGAM include two categories:

- The first category equivalently transforms the LiNGAM into a standard linear Independent Component Analysis (ICA) model (e.g. $\mathbf{X} := (I - B)^{-1}\mathbf{N} \Rightarrow \mathbf{X} := A\mathbf{N}$), and uses corresponding statistical techniques to solve this linear system (Shimizu et al., 2006).
- The second category directly resorts to least squares estimation or maximum likelihood estimation methods to search for the most reasonable causal ordering (Shimizu et al., 2011).

The other typical form is the Additive Noise Models (ANMs) (Hoyer et al., 2008), which constrains the FCMs by the third derivative of nonlinear function f, along with a broader summary(Bühlmann et al., 2014) (namely the Causal Additive Models (CAMs) mentioned early in this section):

$$\boldsymbol{X} := f(\boldsymbol{PA}_X) + \boldsymbol{N}, \tag{4.3}$$

where $\boldsymbol{PA}_X$ and $\boldsymbol{N}$ respectively represent the direct parents of the observed variable vector $\boldsymbol{X}$ and the corresponding (additive) Gaussian or non-Gaussian independent noise that together generate $\boldsymbol{X}$. It's crucial to point out that, both aforementioned CAMs variants (LiNGAM and ANMs) are implicitly complying with the assumptions of Physically Independent Mechanism and Independence of Cause and Mechanism (ICM) mentioned in Section 3.4.4 "*Independence of Cause and Mechanism: Intuition of Causal Asymmetry*", which are thus commonly referred to as the "Independence-Noise-Based approaches" (just meaning the same as "Independence-Mechanism-Based approaches").

4.2 Hybrid-based Approaches for Causal Discovery

In this section, I will informally introduce three hybrid-based causal discovery methodologies originally developed by the DMIR (Data Mining and Information Retrieve) laboratory. Hybrid-based causality approaches are particularly proposed to address the issues in relation to large-scale and high-dimensional causal discovery that traditional methods may fail to approach.

Reminder: Several friendly hooks here helping direct you back to the previous Section 4.1, the next Chapter 5, the current Chapter 4, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

4.2.1 Framework Overview

To begin with a big picture, it may be helpful for one to view the basic algorithmic behaviors adopted by the hybrid-based approaches in generally two steps:

1. Implement a strategically graphical decomposition based on the data generation process assumption mentioned in Section 3.3 and the algorithmic rationale mentioned in Section 4.1.1.
2. Iteratively or repeatedly apply causal inference over each of the subgraph (resulted from the global graphical decomposition) based on the data generation mechanisms assumption mentioned in Section 3.4 and the algorithmic rationale mentioned in Section 4.1.2.

Three representative hybrid-based causality frameworks (SADA (Cai et al., 2013), MLC-LiNGAM (W. Chen et al., 2021a), FRITL (W. Chen et al., 2021b)) are then introduced respectively. Not matter which type I showcase in the following, **the cruxes in such methodologies consist in figuring out or utilizing unique characteristics of when a subgraph forms**. Keeping this in mind helps readers better grasp the functional concept thereof, such as ″causal cuts (Cai et al., 2013)″, ″maximal cliques (W. Chen et al., 2021a)″, and ″Triad conditions (W. Chen et al., 2021b)″ in terminology, that smoothly glues different assumption-based algorithms into a final holistic framework.

4.2.2 The SADA Algorithm

The hybrid-based causal discovery framework referred as to SADA (Scalable cAusation Discovery Algorithm) (Cai et al., 2013) features a strategic search on the ″causal cuts″ capable of partitioning global variables into the different subgraphs that are maximally independent of each other. Such available independency at local subsets is guaranteed by the (common) sparsity of a causal graph. Hence, effectiveness of small-scale causality algorithms is sufficiently harnessed over the subgraphs.

Algorithm 1 Hybrid-based Framework of the SADA Algorithm (Simplified Version)

Input: Data $\boldsymbol{X} = \{x_1, ..., x_d\}$, variable set V_X ($|V_X| = d$), variable threshold θ ($\theta < d$)
Output: causal graph $\mathcal{G}_X$

```
if $|V_X| \leq \theta$ then
    Return $\mathcal{G}_X \leftarrow CausalDiscovery(\boldsymbol{X})$        ▷ Apply causality algorithms mentioned in Section 4.1.2.
end if
$\{C, V_X^1\}, \{C, V_X^2\} \leftarrow GetCausalCut(V_X)$
$\mathcal{G}_X^1 \leftarrow \boldsymbol{SADA}(\boldsymbol{X}, \{C, V_X^2\}, \theta)$
$\mathcal{G}_X^2 \leftarrow \boldsymbol{SADA}(\boldsymbol{X}, \{C, V_X^2\}, \theta)$
Return $\mathcal{G}_X \leftarrow Merge(\mathcal{G}_X^1, \mathcal{G}_X^2)$
```

4.2.3 The MLC-LiNGAM Algorithm

The hybrid-based causal discovery framework referred as to MLC-LiNGAM (Linear non-Gaussian Acyclic Model with Multiple Latent Confounders) (W. Chen et al., 2021a) composes three sequential stages that are "hindsightly" organized through several "specific graphical patterns" at which multiple latent confounders may hint initially. By graphical patterns I mean the subgraphs (referred to as maximal cliques) over which none of the pairwise variable within satisfies conditional independency or the independence noise assumption, which thus suggests a kind of externally confounding dependency that the MLC-LiNGAM is anticipated to detect at the end.

Algorithm 2 Hybrid-based Framework of the MLC-LiNGAM Algorithm (Simplified Version)

```
Input: Data $X = \{x_1, ..., x_d\}$, independence test threshold $\alpha$ ($\alpha < 0.05$)
Output: causal graph $\mathcal{G}_X$

Stage-1: $\mathcal{G}_X^1 \leftarrow CausalDiscovery(X, \alpha)$      ▷ Apply the PC algorithm (Section 4.1.1).
Stage-2: $\mathcal{G}_X^2 \leftarrow CausalDiscovery(X, \mathcal{G}_X^1)$      ▷ Based on the LiNGAM rationale (Section 4.1.2).
Stage-3: $\mathcal{G}_X \leftarrow CausalDiscovery(X, \mathcal{G}_X^2)$      ▷ Apply algorithms iteratively over maximal cliques.
```

Note: Complete and formal algorithmic implementation is available in Appendix B.

4.2.4 The FRITL Algorithm

Akin to the above MLC-LiNGAM algorithm, another hybrid-based framework referred as to FRITL (W. Chen et al., 2021b) composes four causal discovery stages — in particular highlighted with an additional technique that is able to detect latent confounders given the "Triad constraints" (Cai et al., 2019). It's worth noticing that the "Triad condition" is exactly well-fitted to presumably tree-like structures that only consist of a small group of variables, resulting in additional causal identification over subgraphs (as compared to the aforementioned MLC-LiNGAM algorithm). Finally, with the FRITL algorithm asymptotically transforming a Partially Directed Acyclic Graphs (PDAGs) (mentioned in Section 4.1.1) obtained by Stage-1 into a DAG, causality algorithms in favor of low-dimensional circumstances can then be suitably applied to the remaining undetermined relationship (e.g. merely pairwise relationships) among variables in the graph.

Algorithm 3 Hybrid-based Framework of the FRITL Algorithm (Simplified Version)

```
Input: Data $X = \{x_1, ..., x_d\}$, independence test threshold $\alpha$ ($\alpha < 0.05$)
Output: causal graph $\mathcal{G}_X$

Stage-1: $\mathcal{G}_X^1 \leftarrow CausalDiscovery(X, \alpha)$      ▷ Apply the FCI algorithm (Section 4.1.1).
Stage-2: $\mathcal{G}_X^2 \leftarrow CausalDiscovery(X, \mathcal{G}_X^1)$      ▷ Based on the LiNGAM rationale (Section 4.1.2).
Stage-3: $\mathcal{G}_X^3 \leftarrow CausalDiscovery(X, \mathcal{G}_X^2)$      ▷ Apply the Triad condition.
Stage-4: $\mathcal{G}_X \leftarrow CausalDiscovery(X, \mathcal{G}_X^3)$      ▷ Apply algorithms iteratively over maximal cliques.
```

5

Methodology

With supplemental field knowledge, I herein as the first-author reiterate and rephrase my primary undergraduate research work "Non-linear Causal Discovery for Additive Noise Model with Multiple Latent Confounders" (X. Chen et al., 2023b), which serves as an effort to make the content of the research paper more accessible for technical or interested readers.

While the Chapter in this report adopted relatively informal language and flexible structure, it's still written for technical readers — including experts, faculty, and/or employers — who wish to make assessment (regarding research experiences) on my undergraduate research work.

Reminder: Several friendly hooks here helping direct you back to the previous Chapter 4, the next Chapter 6, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

5.1 Overview: A Hybrid-based Framework for Generic Causal Discovery

It is worth pointing out at the beginning that, our work, the ***NonlinearMLC*** (**Non-linear** functions with **M**ultiple **L**atent **C**onfounders) theory and algorithm in causality, relies heavily on the theoretical Causal Assumption imposed upon machine learning, as well as the algorithmic basis provided by both the state-of-the-art approaches and hybrid-based framework in Causal Discovery, which I have introduced in Section 3.3, Section 3.4, Section 4.1, and Section 4.2 in this report, respectively.

This Chapter is organized via commencing with an example that showcases the intuition related to our primary finding in Causal Discovery, then formalizing both the model and theory capable of discovering causation in nonlinear systems with unknown factors (aka. the generic circumstance), and finally offering a solution in form of an out-of-box Causal Discovery algorithm.

An Example: Nonlinear causation identification under the presence of indirected latent confounding, which is illustrated through statistical asymmetry reflective of two hypothetical causal direction.

Model and Theory: The ***NonlinearMLC*** model with its identification theory, which can be viewed as the generalization of the well-know causal ANMs (Hoyer et al., 2008) with the latent confounders.

Solution: An algorithm with its Python implementation (will also be introduced in Section 6.1) integrated by ***CADIMULC*** (will also be introduced in Section 7.1), our associated open source repository, in Github: https://github.com/xuanzhichen/cadimulc/tree/master.

5.2 An Example: Indirected Nonlinear Confounding

Metaphor in Figure 5.1 by itself characterizes the heart of our idea in this work in terms of intuitive causal insights. The rest of the paper or the rest of this Chapter serves as the formalization for the causal identification condition, with which a novel hybrid-based Causal Discovery algorithm can equip to handle generic causation with nonlinearity and multiple latent confounders.

Reminder: Several friendly hooks here helping direct you back to the previous Section 5.1, the next Section 5.3, the current Chapter 5, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

5.2.1 A Standard Deconfounding Process in Inferred Causation

Deconfounding Under Observed (Directed or Indirected) Confounders

By deconfounding I mean the mainstream machine learning strategy by which modern Causal Discovery approaches discover a causal order (relative to the Directed Acyclic Graphs (DAGs)) in an iterative way. For instance, if a set of sharing parents (aka."observed confounders") relative to two variables are observed, the causal relationship between these two variables cannot be inferred — unless the influence of the sharing parents has been removed; such strategy (of removing confounding) can then be iteratively implemented top-down through the DAGs (e.g. searching from the exogenous node at the beginning, and all the way to the leaf node).

Notice that the action of "deconfounding" in part resembles the action of "Conditional Independence Test (CIT)", except that the former realizes its purpose through the regression technique.

Deconfounding Under the Presence of (Indirected) Latent Confounders

Given a hypothetical causal-and-effect relationship (e.g. $C \longrightarrow E$) between two variables (C and E), and discussing the unobserved causal influence (e.g. can be indirected) from one of their parents (denoted either as $\overline{pa}_C$ or $\overline{pa}_E$, as opposed to the observed one pa_C or pa_E) by dividing the circumstances into two diagrams (shown in Fig 5.1, the left panel and the right panel, respectively), let me "hindsightedly" introduce the process through which the Causal Asymmetry is identified.

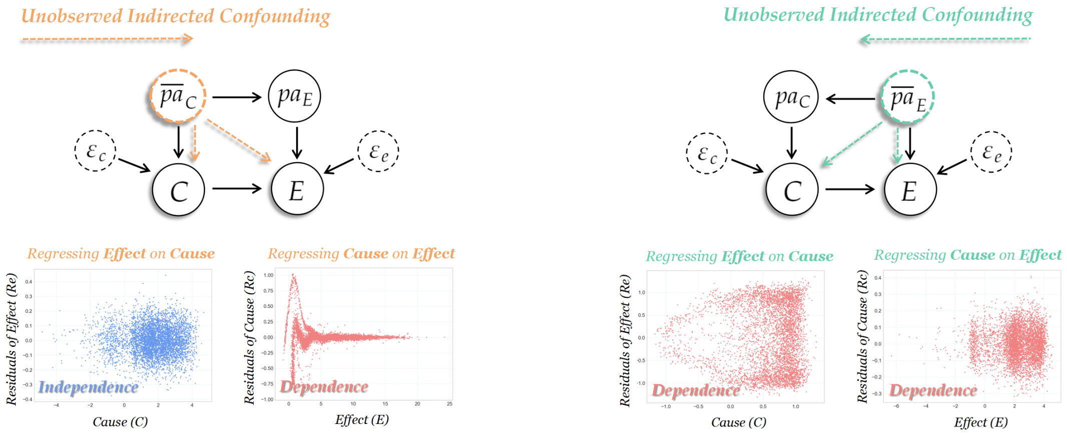


Figure 5.1: *Intuitions of non-linear causal identification under indirected latent confounding. Take the identification over a non-linear additive-noise-model denoted as "cause-and-effect" C $\longrightarrow$ E. Obviously, C $\longrightarrow$ E cannot be methodologically identified, if both of their parent are unobserved (e.g. $\overline{pa}_C$ and $\overline{pa}_E$), amounting to an unobserved common cause.* ***The question is****, whether C $\longrightarrow$ E keeps identifiable if only* ***one side*** *of the parent is unobserved and even triggers the indirected confounding (e.g. C $\longleftarrow pa_C \longleftarrow \overline{pa}_E \longrightarrow$ E)? (Image reprinted from X. Chen et al., 2023b).*

As illustrated by the instance shown above, the "deconfounding" procedure results in the residuals of C and E (denoted as R_C and R_E in Equation 5.1) by regressing on all the hypothetical observed parents respectively (including C and E).

$$< R_C, R_E > = \begin{cases} R_C := C - \hat{f}_C(E,\ pa_E);\ R_E := E - \hat{f}_E(C,\ pa_E), & \text{Left Panel,} \\ R_C := C - \hat{g}_C(E,\ pa_C);\ R_E := E - \hat{g}_E(C,\ pa_C), & \text{Right Panel.} \end{cases} \tag{5.1}$$

Combining the Independence Test, as shown in previous literature (Hoyer et al., 2008; Peters et al., 2017), leads to the statistical asymmetry, indicating that only if (after "deconfounding") the statistical patterns (shown in Equation 5.2) occur simultaneously — independence showcased between R_E and C, whereas the dependence showcased between R_C and E — could we say the (hypothetical) direction of the "cause-and-effect" relationship $C \rightarrow E$ is identifiable from data.

$$(R_E \perp\!\!\!\perp C)\ \wedge\ (R_C \not\perp\!\!\!\perp E). \tag{5.2}$$

In light of this basis, and given two different conditions in relation to the absence of parents ($\overline{pa}_C$ or $\overline{pa}_E$) that raise latent confounding (e.g. light-orange colored arrows that $C \leftarrow \overline{pa}_C \rightarrow pa_E \rightarrow E$ on the left panel, or light-green colored arrows $C \leftarrow pa_C \leftarrow \overline{pa}_E \rightarrow E$ on the right panel), Fig 5.1 argues that only the circumstance shown in the left panel illustrates an identifiable causal relationship (even with latent confounding). In other words, one cannot statistically infer the cause-and-effect relationship (without physical experiments) given the circumstance shown in the right panel.

5.2.2 (Indirected) Latent Confounding, Linearity, and Nonlinearity

"Deconfounding" deserves further discussion since existing work (Maeda et al., 2021; Maeda et al., 2024) on the similar topic — nonlinear Causal Discovery with latent confounders — haven't had an identification condition (in terms of the Causal Graph) akin to the illustration shown in Fig 5.1. Notice that the difference regarding the unobserved patent of which side is trivial in the context of linearity, since linear causal models wouldn't be distorted by indirected confounding.

Specifically, if considering the latent confounding triggered by unobserved parents $\overline{pa}$, the explanation exhibits an intuitive side when we draw comparison with the methodology in linear circumstances: No matter which latent confounder raised by the unobserved parents $\overline{pa}_C$ or $\overline{pa}_E$, the causal information "flow" alongside the indirected confounding path ending up between C and E will be "blocked" by one of their observed parents (e.g. $C \leftarrow pa_C \leftarrow \overline{pa}_E \rightarrow E$ is blocked by pa_C; $C \leftarrow \overline{pa}_C \rightarrow pa_E \rightarrow E$ is blocked by pa_E).

Such blocking behaviors entirely "blocks" the information flow with respect to the confounding path, which means being capable of deconfounding by linear regression methodologically. This strategy, unfortunately, could not pay off in terms of non-linear functions, due to the variables' non-linear interaction compromising the effect of regression. I will continue to explain for this point in Section 5.3.

5.3 Model: Research Motivation & Problem Statement

I herein formalize the issues of nonlinearity through first restricting the Causal Model (I've introduced in Section 3.2 in this report) upon a well-known subclass —Causal Additive Models (CAMs)— capable of implying the Structural Identifiability (I've introduced in Section 4.1.2 in this report), and then modifying the function composition to characterize the effects of multiple latent confounding. Additionally, this section provides the rephrased and supplemental content for my undergraduate research work (X. Chen et al., 2023b) (*in Section 3: Model, Assumption, and Causal Identification Theory*).

Reminder: Several friendly hooks here helping direct you back to the previous Section 5.2, the next Section 5.4, the current Chapter 5, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

5.3.1 Motivation: Indeterministic Parents-Child Causal Relationships

While a deterministic or pseudo-indeterministic system (Spirtes et al., 2000), a model regularly used to represent the causal relationship, cannot satisfy our cases concerning the presence of latent confounders, we found that an indeterministic (as its opposite) does not necessarily become trivial if one could view the role of some latent confounders as same as the unobserved parents.

In a narrow sense — dividing a node's parents (according to Directed Acyclic Graphs (DAGs)) into the observed and the unobserved ones — we are motivated to reconsider another structural model (Equation 5.3) capable of unifying causal properties in both nonlinearity and latent confounding.

Let $\mathcal{G}_X$ the DAGs of variables $\boldsymbol{X} = \{x_1, x_2, \ldots, x_d\}$, with i.i.d. noises $\boldsymbol{\varepsilon} = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_d\}$. Presuming the additive-noise-models (*ANMs*) (Hoyer et al., 2008) — a type of CAMs (I've introduced in Section 4.1.2 in this report) specified in three-differentiable nonlinearity — the Causal Model $< \mathcal{G}_X, \Theta_{\mathcal{G}_X} > = < \mathcal{G}_X, \boldsymbol{\varepsilon}, \boldsymbol{f}_X >$ (I've introduced in Section 3.2 in this report) can be formalized in form of the parents-child generation procedure, with the pairwise relationshiop $x_j \rightarrow x_i$ that

$$x_i \;:=\; \Theta_{\mathcal{G}_X}(\boldsymbol{pa}(x_i) + \overline{\boldsymbol{pa}}(x_i)) \;:=\; \sum_{x_j \in \boldsymbol{pa}_i} f_{ij}(x_j) \;+\; \xi_i. \tag{5.3}$$

We highlight that in Equation 5.3 the " (multiple) latent confounding" from "unobserved parents" $\overline{\boldsymbol{pa}}$ (observed parents $\boldsymbol{pa}$ analogically) is incorporated to the **extensive noises** that $\xi_i := \varepsilon_i \cup \boldsymbol{f}(\overline{\boldsymbol{pa}}_i)$. For the sake of simplicity, we specify the model generated by **Non-linear** functions $\boldsymbol{f}_X$ with **M**ultiple **L**atent **C**onfounders (in ξ_X) as the ***Nonlinear-MLC*** model.

Additionally mild assumptions required in the Causal Discovery task are listed as follows (I've also introduced in Section 3.3.5 in this report): **A-1** Markov Assumption: Independence yielded by $\mathcal{G}_X$ is consistent with ones over distributions P_X. **A-2** Faithfulness Assumption: Distributions P_X faithfully encode independence entailed only by $\mathcal{G}_X$.

5.3.2 Problem: Latent (Causal) Structure and Composite CAMs

In light of the structural formalism of the ***Nonlinear-MLC*** model, we then rephrase our interested problem under the fundamental Causal Discovery framework, namely to discover a latent structure (Pearl, 2009) relative to the Causal Model. By latent (causal) structure we mean a double tuple

$< \mathcal{G}_{X'}, \boldsymbol{X'} >$ where $\mathcal{G}_{X'} \subset \mathcal{G}_X$ and $X' \subset X$. In other words, the ***Nonlinear-MLC*** model establish an omniscient perspective regarding the observationality of child-parent relationships, whereas in practice the task is to discover $\mathcal{G}_{X'}$ over $\boldsymbol{X'}$ with the presence of unobserved parents (latent confounders).

Traditionally, the aforementioned task (Causal Discovery with latent confounding) could be done by expunging causal functions' effects with linear regression, and thus revealing the testable independence noise (aka. the residuals, as Equations 5.1 and 5.2 that I mentioned in Section 5.2.1 in this report). Contrasting with linear combinations (e.g. $\phi(\ \alpha\ (\beta\varepsilon_i)\ ) = (\alpha \cdot \beta) \cdot \varepsilon_i$), the *Nonlinear-MLC* model by Equation(5.3) (in terms of the testable independence noise) **CANNOT** be expanded as:

$$x_i \ := \sum_{\varepsilon_k \in \boldsymbol{\varepsilon} \setminus \{\varepsilon_i\}} \phi(\varepsilon_k) + \varepsilon_i. \tag{5.4}$$

Composite non-linear functions $\phi := f(f(...))$ relative to independent noises **cannot** be accessible, in the sense that "composite *CAMs*" (or equally referred to as the cascading *ANMs* (Qiao et al., 2021)) do not hold with embedded latent noises $\boldsymbol{\varepsilon}_{\overline{\boldsymbol{pa}}}$, leading to the infeasibility to expunge nonlinear effects from within. Once again, contrasting with linear regressor (denoted as $\mathcal{R}(\boldsymbol{pa}_i)$), the presence of "endogenous (unobserved) dependence $\boldsymbol{\varepsilon}_{\overline{\boldsymbol{pa}}}$"will compromise (nonlinear) regression, **FAILING** to satisfy the following deconfounding procedure (I mentioned in Section 5.2.1):

$$\left(x_i \ - \sum_{\varepsilon_k \in \boldsymbol{\varepsilon} \setminus \{\varepsilon_i\}} \phi(\varepsilon_k) \ = \ \varepsilon_i \right) \Rightarrow \ \left(x_i \ - \ \mathcal{R}(\boldsymbol{pa}_i) \ = \varepsilon_i \right). \tag{5.5}$$

Hence, the testable independence noise cannot be yielded due to the embedded latent noises (in terms of unobserved parents) when it comes to nonlinearity, and herein lies the problem.

5.4 Theory: The L-ANMs Identifiable Condition

Aiming at mitigating the above issue, I will commence this section with our proposed Lemma 1, arguing as the (theory for) **latent additive noise models (*L-ANMs*)**, to stipulate a novel identifiable condition for the *Nonlinear-MLC* models (I mentioned in Section 5.3.1). Combining with the previous section, this section provides the rephrased and supplemental content for my undergraduate research work (X. Chen et al., 2023b) (*in Section 3: Model, Assumption, and Causal Identification Theory*).

Reminder: Several friendly hooks here that direct you back to the previous Section 5.3, the next Section 5.5, the current Chapter 5, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

5.4.1 Contribution as a Causal Identifiable Condition

One may view Lemma 1 as an expedient that aims to make the most of the extant Causal Discovery methodologies under the condition of nonlinearity and multiple latent confounding. It's important to notice that Lemma 1 by itself **DOES NOT** imply a kind of new identifiability for extant Causal Discovery approaches. Nevertheless, Lemma 1 severs as a well-defined guidance that can be straightforwardly applied to the deconfounding procedure (I mentioned in Section 5.2.1) in nonlinear cases, which reduces the influence from the embedded latent noises (in terms of unobserved parents) (I mentioned in Section 5.3.2).

Theory in Relation to my Work

Lemma 1 *Assuming data generation procedures are consistent with Equation (5.3), the pair* $C \rightarrow E$ *among other unobserved pairs* $\boldsymbol{C}^* \rightarrow E$ *is identifiable if and only if*

$$(\xi_E \perp\!\!\!\perp C) \wedge (\xi_E := \varepsilon_E \cup f(\boldsymbol{C}^*)) \tag{5.6}$$

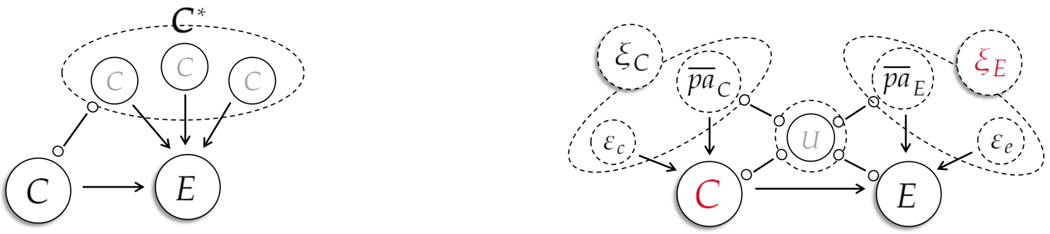
is satisfied, where other multiple unobserved causes $\boldsymbol{C}^*$ *are denoted as* $\boldsymbol{C}^* := \boldsymbol{C} \backslash C = \overline{\boldsymbol{pa}}_E$.

Contributions of the *L-ANMs* Lemma are **three-fold**:

- It is targeted for non-linearity, not necessarily a compulsion for linear cases.
- A succintly deterministic (well-defined) identification condition, compared to notions in previous work (e.g. "C-ANMs" [Qiao et al., 2021] or "CAM-UV" [Maeda et al., 2021]).
- Akin to the well-known identification condition $((\varepsilon_E \perp\!\!\!\perp C) \wedge (\varepsilon_C \perp\!\!\!\perp E \mid C))$ by **Independence Causal Mechanism**(**ICM**) [Peters et al., 2017] (I've mentioned in **??** in this report):
  1. Lemma *L-ANMs* explicitly states a version concerning the identifiable condition in multivariate cases (e.g. $((\varepsilon_E \perp\!\!\!\perp C)$ in bi-variable cases VS. $((\xi_E \perp\!\!\!\perp C)$ in multivariate cases);
  2. Lemma *L-ANMs* permits $(\xi_C \not\perp\!\!\!\perp E \mid C)$, as different to $(\varepsilon_C \perp\!\!\!\perp E \mid C))$, which is essentially how we characterize the latent confounding by unobserved parents in this work.

5.4.2 Graphical Intuition

For illustration, Figure 5.4 (a) pictures our interested "cause-and-effect" circumstance in Causal Graph, where a number of other causes (denoted as C^*) are unobserved (namely the unobserved parents of the effect variable). In light of this, Figure 5.4 (b) is essentially a systematic characterization of the relationships among variables (e.g. the cause, the effect, their parents, and many other variables in the system). Lemma 1 is then figuratively shown in Figure 5.4 (b) (marked as red).



(a) (b)

Figure 5.2: *Graphical intuitions of Lemma 1 (Latent-ANMs) with the symbol "o-o" characterizing the uncertain causal directions "->" or "<-". (a) The structure involving relations between interested cause C and multiple unobserved causes* $\boldsymbol{C}^*$ *given the effect E. (b) The general Nonlinear-MLC model with respect to the structure in (a), where U summarizes the rest of observed variables (Image reprinted from X. Chen et al., 2023b).*

5.4.3 Mathematical Backup

As compared with Equation 5.5, the result of *L-ANMs* is to inspire a seeking for **empirical regressor** $\mathcal{R}_i$ to discover independence between ξ_i and x_j (similar to the binary case ξ_E and C shown in Lemma 1). In other words, Equation 5.3 indicates that, if the search space of machine learning algorithms involves a case in which the residual ξ_i is independent of the variable x_j — after iteratively updating the regressor $\mathcal{R}_i$ (by updating the set of regressing variables x_h), then the pairwise causal relationship $x_j \rightarrow x_i$ can be discovered from data (similar to the binary case $C \rightarrow E$ shown in Lemma 1).

A scratch of proof (about the aforementioned content) with slight algebra in Equation 5.3 shows:

$$x_i - \underbrace{\sum_{x_h \in \boldsymbol{pa}_i \setminus \{x_j\}} f_{ih}(x_h)}_{\mathcal{R}_i} = f(x_j) + \underbrace{\left(\sum_{x_k \in \overline{\boldsymbol{pa}}_i} f_{ik}(x_k) + \varepsilon_i \right)}_{\xi_i}. \tag{5.7}$$

In depth, hypothesizing the causal generative mechanism (Equation 5.3) for $C \longrightarrow E$ as model $\mathcal{M}_1$ and the reverse one $E \longrightarrow C$ as model $\mathcal{M}_2$, along with the likelihood functions $\mathcal{L} = \log P(\cdot)$, the rest of the proof can be reduced as the following algebraic asymmetry (Hoyer et al., 2008):

$$\frac{\partial}{\partial x_j} \left(\frac{\partial^2 \mathcal{L}(\mathcal{M}) / \partial x_j^2}{\partial^2 \mathcal{L}(\mathcal{M}) / \partial x_i \partial x_j} \right) \begin{cases} \neq 0, & \mathcal{M} = \mathcal{M}_1 \quad \text{(causal direction } C \longrightarrow E\text{)}, \\ = 0, & \mathcal{M} = \mathcal{M}_2 \quad \text{(anti-causal direction } E \longrightarrow C\text{)}. \end{cases} \tag{5.8}$$

That is, the condition in Lemma 1 ensures a compatible proof framework, even we presume the influence of latent variables, leading to the fact that the *Nonlinear-MLC* model only holds in the causal direction and can not be inverted. Technical readers may refer to Appendix A in this report for the complete proof with more mathematical details (if necessary).

5.5 Solution: The Nonlinear-MLC Algorithm

When I was writing the paper (X. Chen et al., 2023b), I was meant to organize the structure in a way commonly adopted by causal discovery related articles to introduce my work. That is, a newly proposed causal discovery algorithm is expected to equip with a causality identifiable theory within. Given our case, the identifiability condition exactly refers to Lemma 1 I just mentioned in the previous section. However, equipping our practical algorithm with associated theorem leads to another question: the *Nonlinear-MLC* models to which Lemma 1 is tied are established in an omniscient perspective regarding the "observationality of child-parent relationships", whereas the algorithm in practice is coping with circumstances with existing "unobserved parents" (latent confounders).

I wish my aforementioned statement hereby fully implies the purpose with which I am writing this section (or *Section 4, "A Theory-based Novel Algorithm for Causal Discovery"* in my related research paper (X. Chen et al., 2023b)): to contribute a Corollary derived from Lemma 1, which enables practical algorithms to asymptotically reach the theoretical causal identification (guaranteed by Lemma 1) amidst the learning environment with nonlinearity and latent confounding.

Reminder: Several friendly hooks here that direct you back to the previous Section 5.4, the next Section 5.6, the current Chapter 5, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

5.5.1 Connecting the Theory with a Practical Graphical Pattern

First, due to the presence of unobserved parents in applications, existing algorithms [Maeda et al., 2021][Tashiro et al., 2014] may end up being stuck in undetermined dependence of variable subsets. The "undetermined dependence" within variable subsets, from the graphical perspective, might indicate the "undirected connection" within maximal cliques $\mathcal{M}$. (Notice that the maximal clique refers to a special pattern of Undirected Graphs, where every variable consisting of the graph is connected with each other without directionality.) Second, one might notice that, in the view of algorithms amidst their computing memory, the undetermined child-parent relationship amounts to the results caused by the existence of either **the truly multiple unobserved parents**, or the **observational variables whose "parental roles" just have not been determined**, and thus have not yet been stored into the algorithms' memory. While the former issue (real unobserved parents) is something we can not currently solve in this work, the later issue (temporarily undermined parents, or personally I would like to refer to it as the ”pseudo-unobserved parents”) is what we aim to fix through strategically adjusting the algorithmic behaviors (e.g. regression) during causal discovery.

For illustration, let's recall the left panel in Figure 5.4. Notice that if there exists local structures or subgraphs $\boldsymbol{\mathcal{M}}$ including C , E, and $\boldsymbol{C}^*$ during a certain stage of causal discovery algorithms, such structures would limit the anticipant independence required by Lemma 1 between $\boldsymbol{C}^*$ and C. (That's because every variable in $\boldsymbol{\mathcal{M}}$ has showcased dependence with each other!) Thus, we propose the following corollary drawing on **empirical regressor** $\mathcal{R}$ to counteract that dependence.

Corollary 1 *Assuming data generation procedures are consistent with Equation (5.3), the pairwise cause-and-effect $C \to E$ over a* ***maximal clique*** $\boldsymbol{\mathcal{M}}$ *is identifiable if and only if*

$$(E - \mathcal{R}_E(\mathcal{M}^*) \perp\!\!\!\perp C) \;\wedge\; (\mathcal{M}^* := \mathcal{M}_{C,E} \setminus \{E\}) \tag{5.9}$$

is satisfied, where $\mathcal{M}_{C,E}$ represents all observed variables including C and E within a maximal clique.

Were I to make Equation 5.9 in Corollary 1 more clear, one can view the term $E - \mathcal{R}_E(\mathcal{M}^*)$ simply as the asymptotic approximation to the term ξ_E in Equation 5.6, Lemma 1.

$$\{E - \mathcal{R}_E(\mathcal{M}^*)\} \;\mapsto\; \xi_E. \tag{5.10}$$

Formal proof related to Corollary 1 can be found in Appendix A in this report. **For readers who want to straightforwardly apply the Corollary** without diving into technical details in Equation 5.9, Figure 5.3 respectively illustrates three of the circumstances over which the maximal-clique-based causal discovery is applied, in the view of algorithms amidst their computing memory. Notice that $\overline{pa}_C$ or $\overline{pa}_E$ denotes the (aforementioned) truly multiple unobserved parents, while the variables U represent the (aforementioned) observational variables whose "parental roles" just have not been determined. Thus, external causal influence from the former is marked as dotted yellow lines; controllable causal influence from the later is on blue with its directionality hypothesized.

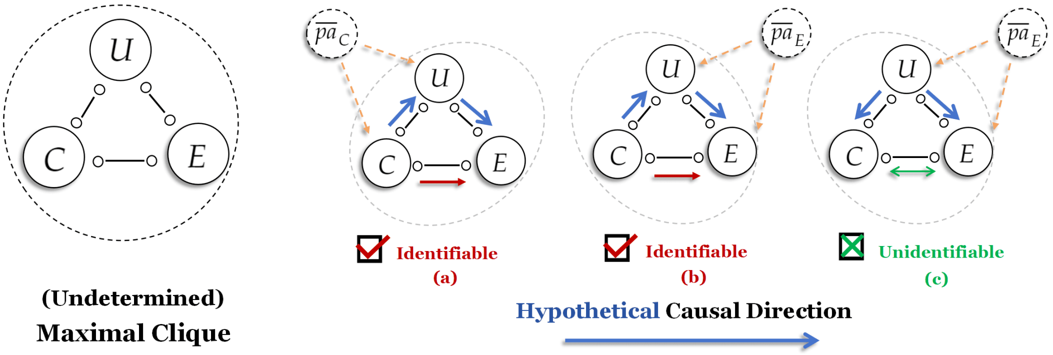


Figure 5.3: *A toy graphical structure, namely $\mathcal{M}_{C,E} = \{C, E, U\}$, illustrates how to determine non-linear identifiability under latent confounding ($\overline{pa}_C$ or $\overline{pa}_E$) by applying Corollary 1. (Image reprinted from X. Chen et al., 2023b).*

For instance, let's take case (c) in Figure 5.3. According to Equation 5.9 in Corollary 1, the term:

$$E - \mathcal{R}_E(\mathcal{M}^*) = E - \mathcal{R}_E(\mathcal{M}_{C,E} \backslash \{E\}) = E - \mathcal{R}_E(C, U) \tag{5.11}$$

informs our algorithm to perform nonlinear regression on variable E through expunging the nonlinear effects from variables C and U. If the regression is successfully done, it will graphically amount to cutting off the dependence "C o-o E" and "U o-o E" from the graph shown in case (c).

Then, the next step is check whether such residuals $R_E = E - \mathcal{R}_E(\mathcal{M}^*)$ are independent of the hypothetical cause C. Looking closely into case (c), notice that even if the dependence "C o-o E" and "U o-o E" are removed, E and C are still correlated due to the unobserved parents $\overline{pa}_E$ that yields latent confounding propagating through the path $C \leftarrow U \leftarrow \overline{pa}_E \rightarrow E$. Hence, our algorithm estimates that, under circumstance (c), nonlinear causal relationship $C \rightarrow E$ cannot be identified (e.g. denoted as "$\leftrightarrow$") from observational raw data, implying that there exists a kind of latent confounding that precludes Corollary 1 from satisfaction.

Repeat my operation above but conduct it in case (a), this time reader will find that the independency required by Corollary 1 is satisfied. The only thing I may need to remind is that, in case (b), related independency will be derived from a "V structure" forming in the graph. This involves the notions of D-Separation that I've mentioned in Section 3.3.4 "*Structural Insights with D-Separation*".

5.5.2 A Two-stage Algorithmic Framework

The hybrid-based framework featured by the *NonlinearMLC* algorithm in our work (X. Chen et al., 2023b) can be easy to understand if one grasps the basic idea entailed by the Corollary mentioned above. Concretely, we apply the novel identification (implied by Corollary 1) on the maximal cliques $\boldsymbol{\mathcal{M}}$ partitioned over causal skeleton $\mathcal{S}_{X'}$, which stands on the basic ground provided by the PC algorithm [Spirtes et al., 2000], along with the algorithm's identification guaranteed by Lemma 2.

Lemma 2 *Suppose that assumptions A1 and A2 hold, every true adjacency pair of variables x_i and x_j in $\mathcal{G}_X$ is in accord with the estimated adjacency pair in causal skeleton $\mathcal{S}_{X'}$ of $\mathcal{G}_{X'}$ using PC algorithm.*

The two-stage hybrid-based framework of the *NonlinearMLC* algorithm (simplified version) is listed:

Algorithm 4 Nonlinear-MLC Algorithm

```
Input: Data $\boldsymbol{X'} = \{x_1, ..., x_m\} (m < d)$,  significant level $\alpha$
Output: Estimated causal graph $\hat{\mathcal{G}_{X'}}$
$\hat{\mathcal{S}_{X'}}, \hat{\mathcal{G}_{X'}} \leftarrow stage1CausalDiscovery(\boldsymbol{X'},\ \alpha)$,
search ← True;
while search do
    $\hat{\mathcal{G}_{X'}} \leftarrow stage2CausalDiscovery(\boldsymbol{X'},\ \alpha,\ \boldsymbol{\mathcal{M}}(\hat{\mathcal{S}_{X'}}),\ \hat{\mathcal{G}_{X'}})$, ▷ Causal inference based on Corollary 1.
    search ← False;
    if $determinedNewDirections(\hat{\mathcal{G}_{X'}})$ then
        search ← True;
    end if
end while
return($\hat{\mathcal{G}_{X'}}$)
```

For Python implementation of the algorithm mentioned above, readers may also move to my related code snippet Listing 6.3 displayed in the next Chapter 6 *"Programming (Code Samples)"*.

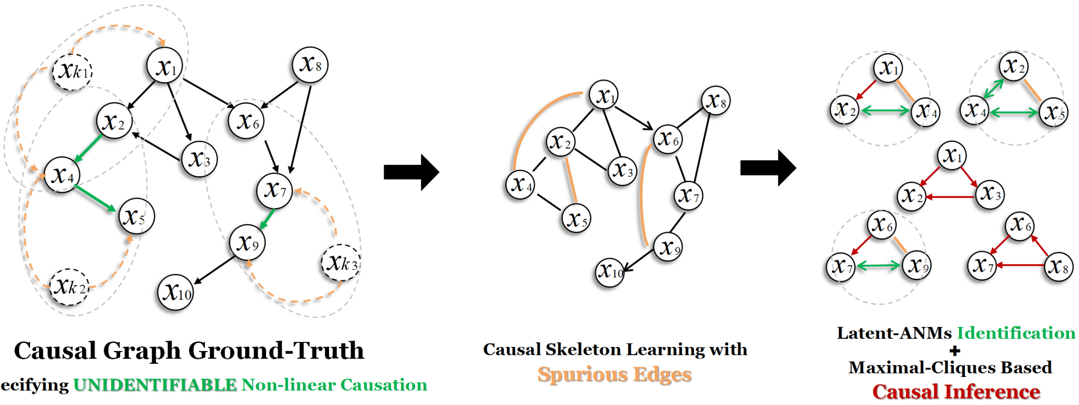

x_{k1}
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
x_10
x_{k2}
x_{k3}
Causal Graph Ground-Truth
Specifying UNIDENTIFIABLE Non-linear Causation
Causal Skeleton Learning with
Spurious Edges
Latent-ANMs Identification
+
Maximal-Cliques Based
Causal Inference

Figure 5.4: *A two-steps method with the spurious edges detecting latent confounders (Image reprinted from X. Chen et al., 2023b).*

According to causality insights borrowed from W. Chen et al., 2021a, another finding in Figure 5.4 lies in the spurious edges (marked in yellow)raised by latent confounders are accompanying with the stage-1 causal skeleton discovery (by the PC algorithm), which will be consistent to the (partial determined) maximal cliques that are comprised of a least one undetermined edge (e.g. ↔ marked in green) after applying Corollary 1 mentioned in the previous section. Therefore, this is the procedure in which the *NonlinearMLC* algorithm conducts nonlinear causal discovery and ultimately detects the presence of latent confounders.

5.6 Summary

Reminder: Several friendly hooks here that direct you back to the previous Section 5.5, the next Chapter 6, the current Chapter 5, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

Leveraging the popular methodology, namely the regression and the independence test, procedure of causal identification after the standard "deconfounding" process (e.g. deconfounding the influence from observed confounders) has been successfully applied in multivariate causal discovery. In practice, nevertheless, only a subset of variables relative to a reasonable system could be measured, leading to the presence of (multiple) unobserved confounders. Additionally, linear causal discovery methodologies capable of expunging the effects of latent confounding through multivariate regression, unfortunately, could not pay off in terms of (regressing) non-linear functions. Both nonlinearity and the existence of latent confounders pose significant challenges on distinguishing causal directions in such generic circumstances.

Our work shown in this report, however, still endeavour to contribute a leeway for appropriately identify causal direction from the data that involves non-linearity and latent confounding. Briefly, our finding (as illustrated in Fig 5.2) consists in that the direction of "cause-and-effect" $C \rightarrow E$ is able to continuously keep identifiable, only if the confounding is triggered by the unobserved parent $\overline{pa}_C$ rather than $\overline{pa}_E$. Equipping with the theoretical conclusion (shown in Section 5.4) , we further proposed a "hybrid" causal discovery algorithm (shown in Section 5.5)that will wisely utilize the regression-independence-test methodology to reach the practical efficiency.

6

Programming (Code Samples)

Through hiding some low-level implementation details for readers, I will introduce the most relevant programming (in Python) of my undergraduate research work (X. Chen et al., 2023b) in this Chapter — in form of attaching my annotations for several separated pieces of the ”quasi-source-code”. The coding display herein covers a typical procedure of the task of causal discovery, such as data generation, automation of causal inference, and graph visualization and evaluation. The source code used for comment in this Chapter depends on the program’s initial version (May, 2024), with its complete programming in Github: https://github.com/xuanzhichen/cadimulc/tree/master.

While the Chapter in this report adopted relatively informal language and flexible structure, it’s still written for technical readers — including experts, faculty, and/or employers — who wish to make assessment (regarding professional coding skills) on my undergraduate research work.

Reminder: Several friendly hooks here that direct you back to the previous Chapter 5, the next Chapter 7, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

6.1 Implementation for the Proposed Algorithm

Given our proposed *NonlinearMLC* algorithm (Section 5.5), this section is meant to showcase only a ”glimpse” of its composition and organization from the perspective of (the simplified) source code.

6.1.1 Module-1: Hybrid Framework Base

```python
class HybridFrameworkBase(metaclass=ABCMeta):
    def __init__(self, _skeleton: ndarray = None, _adjacency_matrix: ndarray = None):
        # parameters setting

    @abstractmethod
    def fit(self, dataset: ndarray) -> ndarray:
        # fitting data via hybrid-based causal discovery
        return self.adjacency_matrix_

    def _causal_skeleton_learning(self, dataset: ndarray) -> ndarray:
        # learning an undirected graph through conditional independence tests
        return self.skeleton_
```

Listing 6.1: *Hybrid Framework Base in Python.*

My Annotation: A hybrid-based causal discovery framework (`HybridFrameworkBase`) refers to the established first-stage of causal skeleton discovery (`_causal_skeleton_learning`) by the Peter-Clark algorithm (Spirtes et al., 2000). `fit` is a required method fitting data via the algorithm, leading to a causal graph in form of an adjacency (Boolean) matrix (`adjacency_matrix_`).

Additional Notes: The framework is also incorporated into the initial stage of the other popular hybrid-based approach (in IEEE-TNNLS, 2021), the MLC-LiNGAM algorithm (W. Chen et al., 2021a), with its complete Python implementation available in Appendix B.

6.1.2 Module-2: Auxiliary Manager for Structure Learning

```python
class GraphPatternManager(object):
    def __init__(
            self,
            init_graph: ndarray,
            managing_adjacency_matrix: ndarray | None = None,
    ):
        self._managing_skeleton = copy(init_graph)
        # other parameters setting

    def identify_directed_causal_pair(self, determined_pairs):
        if len(determined_pairs) > 0:
            # update self.managing_adjacency_matrix
        return self

    def get_undetermined_cliques(self, maximal_cliques: list[list]) -> list:
        maximal_cliques_undetermined = contrasted_search(
        self._managing_skeleton
        self.managing_adjacency_matrix,
        )
        # Mark undetermined cliques (with respect to the original maximal cliques)
        # if there is at least one edge (in a clique) remaining undetermined.
        return maximal_cliques_undetermined

    @staticmethod
    def recognize_maximal_cliques_pattern(
            causal_skeleton: ndarray,
    ) -> list[list]:
        maximal_cliques = []
        # Search (original) maximal cliques over the causal skeleton
        return maximal_cliques
```

Listing 6.2: *Auxiliary Manager (Graphical Pattern Manager) in Python.*

My Annotation: An auxiliary module (`GraphPatternManager`) embedded in algorithms assist the algorithmic behavior in graphical pattern recognition (`recognize_maximal_cliques_pattern`) (I've mentioned in Section 5.5.1 "*Connecting the Theory with a Practical Graphical Pattern*" in this report). With the help of `get_undetermined_cliques` and `identify_directed_causal_pair`, the module as well manages (transitional) adjacency matrices amidst the procedure between (initial) causal skeleton learning and (final) causal direction orientation — namely an estimated graph is iteratively transferred from the state of undetermined `init_graph` to the determined `managing_adjacency_matrix`.

6.1.3 Module-3: Primary Programming of the Algorithm

It's worth noticing that the following code sample in this section is echoing to the pseudo-algorithm mentioned in Section 5.5.2 *"A Two-stage Algorithmic Framework"*. Meanwhile, the code sample within is simplified for the sake of readability. Complete Python source code is available at:

https://github.com/xuanzhichen/cadimulc/blob/master/cadimulc/hybrid_algorithms/

```
### Input: Data (dataset), significant level (pc_alpha)
### Output: Estimated causal graph (in form of adjacency matrix)

class NonlinearMLC(HybridFrameworkBase):
    def __init__(
            self,
            ind_test: str = 'kernel_ci',
            pc_alpha: float = 0.05,
            _regressor: object = GAM() # (nonlinear) Generalized Additive Models
    ):
        HybridFrameworkBase.__init__(self, pc_alpha=pc_alpha)
        # other parameters setting

    def fit(self, dataset: ndarray) -> ndarray:
        ### Stage-1 Causal Discovery
        # Reconstruct a causal skeleton using the PC-stable algorithm.
        causal skeleton = self._causal_skeleton_learning(dataset)

        # Initialize a graph pattern manager for subsequent learning.
        graph_pattern_manager = GraphPatternManager(
            init_graph=causal skeleton
        )

        continue_search = True
        while continue_search:

            ### Stage-2 Causal Discovery
            # Obtain the cliques that remain at least one edge undetermined.
            undetermined_maximal_cliques = (
                graph_pattern_manager.get_undetermined_cliques(maximal_cliques)
            )

            # End if all edges over the cliques have been determined.
            if len(undetermined_maximal_cliques) == 0:
                break

            # Perform the (main part of) the NonlinearMLC causal discovery.

            # !!! Noice !!!
            # See the method "_clique_based_causal_inference" in the next page.
            determined_pairs = self._clique_based_causal_inference(
                undetermined_maximal_cliques=undetermined_maximal_cliques
            )

            # Orient determined causal directions after a search round over maximal cliques.
            graph_pattern_manager.identify_directed_causal_pair(
                determined_pairs=determined_pairs
            )

            # Update the causal adjacency matrix after a search round over maximal cliques.
            self._adjacency_matrix = (
                graph_pattern_manager.managing_adjacency_matrix
            )

            # Check if new causal relations have been determined after the last round searching.

            # !!! Noice !!!
            # For simplicity, I didn't display this minor method in the previous section "Module-2".
            newly_determined = (
                graph_pattern_manager.check_newly_determined(
                    undetermined_maximal_cliques
                )
            )

            # End if there is none of new causal relation advancing the further search.
            if not newly_determined:
                continue_search = False

        return self.adjacency_matrix_

    def _clique_based_causal_inference(self, maximal_cliques: list[list]) -> list:
        """
        Parameter Required: methods related to
            (i)  nonlinear regression
            (ii) (conditional) independence tests

        # !!! Notice !!!
        # This method implements a strategic regression-independence methodology for causal inference,
        # which may be too technical to be displayed in an introductory report.

        # !!! Notice !!!
        # Interested technical readers seeking thorough details related to this method
        # may refer to my paer and github source code at
        # (line 623, hybrid_algorithms.py, initial version 0.0.0).
        """

        ### Implementation Ideas:
        # In light of the L-ANMs theory (proposed in Section 5.4 and 5.5 in this report),
        # start the search round by conducting non-linear causal inference based on maximal cliques.

        return maximal_cliques_undetermined

```

Listing 6.3: *Primary Programming of the NonlinearMLC Algorithm in Python.*

My Annotation: The primary programming of the *NonlinearMLC* algorithm (NonlinearMLC) operates as an incorporation of the constraint-based (Section 4.1.1) and the functional-based (Section 4.1.2) causal discovery methodology, aiming at approaching the generic causal inference over nonlinear data with the presence of multiple unknown factors. Accordingly, NonlinearMLC thereby features its algorithmic capability of exploiting the (asymptotic approximation of) nonlinear causal identification with multiple latent confounders, which is also proposed as the Latent-ANMs causal identification (I've highlighted in Section 5.4 and Section 5.5 in this report as the main contribution of my undergraduate research work (X. Chen et al., 2023b)).

Based on the code snippet Listing 6.3, several programming details are explained as followings:

- `HybridFrameworkBase`: The framework base (mentioned in Section 6.1.1) acts as Class inheritance relative to the *NonlinearMLC* algorithm, in order to conduct first-stage skeleton discovery by the PC algorithm (Spirtes et al., 2000) during the entire hybrid-based causal discovery.
- `__init__`:
  - `ind_test` refers to the recommended popular non-linear independence-tests method: Kernel-based Conditional Independence tests (KCI) (Zhang et al., 2012). Aside from `kci`, the other parameter setting `hsic`, Hilbert-Schmidt Independence Criterion (for General Additive Models (HSIC-GAMs)) (Gretton et al., 2005), is also available.
  - `pc_alpha` refers to the significance level of independence tests (aka. P-value), which is required by the constraint-based methodology incorporated in the initial stage of the hybrid causal discovery framework.
  - `_regressor=GAM()` refers to the nonlinear regression method adopted by the algorithm in subsequent stages, which is well-fitted to perform regression over (nonlinear) additive models (Wood, 2004) — the identifiable functional class I mentioned in Section 4.1.2.
- `fit`: Fitting data via the *Nonlinear-MLC* causal discovery algorithm.
  - The procedure comprises the causal skeleton learning in the initial stage, along with the causal identification procedure involving non-linear regression and independence tests for the subsequence.
  - Following the well-known divide-and-conquer strategy, non-linear causal inference are conducted over the maximal cliques recognized from the estimated causal skeleton.
- `_clique_based_causal_inference`:
  - For each of the undetermined maximal cliques (e.g. at least one edge within a maximal clique remains undirected) with respect to the whole maximal-clique patterns, the algorithm conducts non-linear regression and independence tests with the additional explanatory variables selected from the undetermined maximal clique.
  - This strategy is argued to enhance the efficiency and robustness as to the non-linear causal discovery with multiple latent confounders, serving as the essence of the *Nonlinear-MLC* algorithm (See ”Latent-ANMs Lemma” for relevant details my paper (X. Chen et al., 2023b), Section 3; or Section 5.5 in this report).

6.2 Module Tests

In light of the three modules (mentioned in the previous section) of our proposed algorithm, this section further provides an overview of their feasibility test respectively.

I should remind herein that each module test may rely on several pre-declared functions with which the module tests are assisted. I try to name these functions clearly so that readers may not need to refer to their concrete definition while glancing over the code sample. By the same token, code sample introduced in this section is processed with simplicity. Complete testing is available at:
https://github.com/xuanzhichen/cadimulc/blob/master/tests/test_hybrid_algorithms.py

Reminder: Several friendly hooks here helping direct you back to the previous Section 6.1, the next Section 6.3, the current Chapter 6, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

6.2.1 Test for Module One

The following code snippet is testing for the correct initialization of `HybridFrameworkBase`, and the foremost encapsulation of the method `get_skeleton_from_pc`, meaning to discover a causal skeleton via the PC algorithm (Spirtes et al., 2000) in the first stage during the entire hybrid-based causal discovery framework.

```python
def test_0x1_causal_skeleton_learning():
    random_seed = 42

    # Randomly generate simulated causal model in a general setting.
    ground_truth, data = simulate_single_case_causal_model(
        linear_setting=False,
        random_seed=random_seed
    )

    display_test_section_symbols()

    print("* Ground Truth Adjacency Matrix: \n", ground_truth)

    # Test the correct initialization of HybridFrameworkBase.

    nonlinear_mlc = NonlinearMLC()
    nonlinear_mlc._dataset = data
    model = nonlinear_mlc

    # Test the foremost encapsulation of get_skeleton_from_pc.
    model._causal_skeleton_learning(data)

    draw_graph_from_ndarray(
        array=ground_truth,
        testing_text='ground_truth'
    )
    draw_graph_from_ndarray(
        array=model.skeleton_,
        testing_text='estimation_skeleton'
    )

    draw_graph_from_ndarray(
        array=model.skeleton_,
        graph_type=True,
        testing_text='estimation_adjacency_matrix'
    )

    plt.show()

    print("* Stage-1 Running Time: ", model.stage1_time_)
```

6.2.2 Test for Module Two

The following code snippet is testing for the Class-methods as to the cliques pattern management (GraphPatternManager). Specifically, the code tests (1) recognizing the maximal-cliques pattern over a causal skeleton; and (2) get undetermined cliques over a partial causal skeleton.

```
def test_0x2_graph_pattern_manager():
    random_seed = 42

    # Randomly generate simulated causal model in a general setting.
    ground_truth, data = simulate_single_case_causal_model(
        graph_node_num=5,
        random_seed=random_seed
    )

     # Obtain the causal skeleton from ground truth.
    causal_skeleton = get_skeleton_from_adjmat(adjacency_matrix=ground_truth)

     # ==================== RECOGNIZE MAXIMAL CLIQUES ========================

    display_test_section_symbols(testing_mark='recognize_maximal_cliques')

    maximal_cliques = GraphPatternManager.recognize_maximal_cliques_pattern(
        causal_skeleton=causal_skeleton
    )

    print(
        "* Maximal Cliques Pattern: ",
        adjust_nested_list_counting_from_one(maximal_cliques)
    )

    # Display the associating causal skeleton.
    draw_graph_from_ndarray(
        array=causal_skeleton,
        testing_text="skeleton"
    )
    plt.show()

    # ============================ GET UNDETERMINED CLIQUES ===============================

    display_test_section_symbols(testing_mark='get_undetermined_cliques')

    # Initialize a graph pattern manager for subsequent tests.
    graph_pattern_manager = GraphPatternManager(init_graph=causal_skeleton)

    # Partially orient the causal skeleton based on existing maximal cliques (random_seed=42).
    determined_clique = maximal_cliques[0] if len(maximal_cliques) > 0 else []
    for i in determined_clique:
        for j in determined_clique[i:]:
            # Partially orient the causal skeleton
            # (notice: avoid acyclic graphs).
            graph_pattern_manager.managing_adjacency_matrix[i][j] = 1
            graph_pattern_manager.managing_adjacency_matrix[j][i] = 0

    # Display the undetermined cliques searching result.
    undetermined_cliques = graph_pattern_manager.get_undetermined_cliques(
        maximal_cliques=maximal_cliques
    )

    print(
        "* Undetermined Cliques: ",
        adjust_nested_list_counting_from_one(undetermined_cliques)
    )

    # Display the associating partial skeleton (adjacency matrix).
    draw_graph_from_ndarray(
        array=graph_pattern_manager.managing_adjacency_matrix,
        testing_text="partial_skeleton"
    )
    plt.show()
```

6.2.3 Test for Module Three (Core Test)

Algorithmic Behavior and Its Encapsulation

The following code snippet is testing for exposing details amidst the fitting procedure of NonlinearMLC ahead of its encapsulation. Specifically, the code tests the data flow in clique-based causal inference (_clique_based_causal_inference).

```
def test_0x3_procedure_fitting():
    random_seed = 42
    # ================= DATA GENERATION AND GROUND-TRUTH PREPARATION ====================

    # Randomly generate simulated causal models in a general setting.
    # e.g. hybrid non-linear and Gaussian noise setting
    ground_truth, dataset = simulate_single_case_causal_model(
        linear_setting=False,
        random_seed=random_seed
    )

    # Initialize nonlinear-mlc ahead of fitting procedure.
    nonlinear_mlc = NonlinearMLC()

    nonlinear_mlc._dataset = dataset
    nonlinear_mlc._causal_skeleton_learning(dataset)

    display_test_section_symbols(testing_mark='corresponding_causal_skeleton')


    draw_graph_from_ndarray(
        array=ground_truth,
        testing_text='ground_truth'
    )
    draw_graph_from_ndarray(
        array=nonlinear_mlc._skeleton,
        testing_text='causal_skeleton'
    )
    plt.show()

    display_test_section_symbols(testing_mark='corresponding_causal_discovery')

    # ========================= LIST STRUCTURAL PROCEDURE CLIPS ============================

    # ----------------------- SETUP CLIQUE-BASED INFERENCE FRAMEWORK ----------------------

    # Recognize the maximal-clique pattern based on the causal skeleton.
    maximal_cliques = GraphPatternManager.recognize_maximal_cliques_pattern(
        causal_skeleton=nonlinear_mlc._skeleton
    )

    # Initialize a graph pattern manager for subsequent learning.
    graph_pattern_manager = GraphPatternManager(
        init_graph=nonlinear_mlc._skeleton
    )

    print(
        "* Whole Patterns of Maximal Cliques: ",
        adjust_nested_list_counting_from_one(maximal_cliques)
    )

    print("* Structural Procedure of Clique-Based Inference Starts...")

    # Perform the nonlinear-mlc causal discovery.
    continue_search = True
    search_round = 0
    while continue_search:

        search_round += 1
        print("* Search Round: ", search_round)

        # Obtain cliques that remain at least one edge undetermined.
        undetermined_maximal_cliques = (
            graph_pattern_manager.get_undetermined_cliques(maximal_cliques)
        )

        print(
            "    * Undetermined Maximal Cliques: ",
            adjust_nested_list_counting_from_one(undetermined_maximal_cliques)
        )

        # End if all edges over the cliques have been determined.
        if len(undetermined_maximal_cliques) == 0:
            print("* End the Searching Round\n")
            break

        # ----------------------- DIVE INTO CLIQUE-BASED INFERENCE  -----------------------
            # !!! Notice !!!
            # Source code within this scope tests
            # a strategic regression-independence methodology for causal inference,
            # which may be too technical to be displayed in an introductory report.

            # !!! Notice !!!
            # Interested technical readers seeking thorough details related to this method
            # may refer to my paer and github source code at
            # (line 1678, test_hybrid_algorithms.py, initial version 0.0.0).
        # ------------------------ DIVE OUT CLIQUE-BASED INFERENCE ------------------------

        print(
            "* Update Current Determined Paris: ",
            adjust_nested_list_counting_from_one(determined_pairs)
        )

        # Orient the determined causal directions
        # after a search round over maximal cliques.
        graph_pattern_manager.identify_directed_causal_pair(
            determined_pairs=determined_pairs
        )

        # Update the causal adjacency matrix and parent-relations set
        # after a search round over maximal cliques.
        nonlinear_mlc._adjacency_matrix = (
            graph_pattern_manager.managing_adjacency_matrix
        )
        nonlinear_mlc._parents_set = (
            graph_pattern_manager.managing_parents_set
        )

        # Display the change related to the adjacency matrix immediately.
        draw_graph_from_ndarray(
            array=nonlinear_mlc._adjacency_matrix,
            testing_text='partial_adjacency_matrix'
        )

        # Check if new causal relations have been determined
        # after the last round searching
        newly_determined = (
            graph_pattern_manager.check_newly_determined(
                last_undetermined_cliques=undetermined_maximal_cliques
            )
        )

        print("* Newly Determined: ", newly_determined)

        # End if none of new causal relation advancing the further search.
        if not newly_determined:
            continue_search = False
            print("* End the Searching Round\n")
        else:
            print("* Continue the Next Searching Round\n")
```

Data Generation, Causal Discovery, and Evaluation

The other code snippet is testing for numerical and empirical results of the fitting procedure, namely the average performance (evaluated by the well-known F1 score) of the algorithm on the general setting with 100 times running (Note: APIs within will be introduced in the next section).

```python
def test_0x4_performance_fitting():
    display_test_section_symbols(testing_mark='repetitive_cases')

    i = 0
    running_times = 100
    step = 50 # increasing the space between random seeds to get a general result
    f1_score_avg = 0
    f1_score_list = []

    while i < running_times:
        try:
            random_seed = copy_and_rename(i + step)
            np.random.seed(random_seed)
            random.seed(random_seed)

            i += 1

            # Randomly simulate causal model in a general case: Non-linear and Gaussian noise.
            generator = Generator(
                graph_node_num=8,
                sample=1000,
                causal_model='hybrid_nonlinear',
                sparsity=0.5
            )
            ground_truth, data = (
                generator.run_generation_procedure().unpack()
            )

            # Perform mlc-lingam causal discovery.
            nonlinear_mlc = NonlinearMLC()
            nonlinear_mlc.fit(data)

            # Conduct causal graph evaluation by F1 score.
            f1_score = Evaluator.f1_score_pairwise(
                true_graph=ground_truth,
                est_graph=nonlinear_mlc._adjacency_matrix
            )
            f1_score_avg += f1_score
            f1_score_list.append(f1_score)

            print("* Case-{}: Pass with F1-Score: {}".
                  format(i, f1_score))

        except Exception as err_msg:
            print("* Case-{}: An Error Occurred: {}".
                  format(i, err_msg))

    print("* Average F-1 Score: ", f1_score_avg / running_times)
    print("* Medium  F-1 Score: ", get_medium_num(f1_score_list))
```

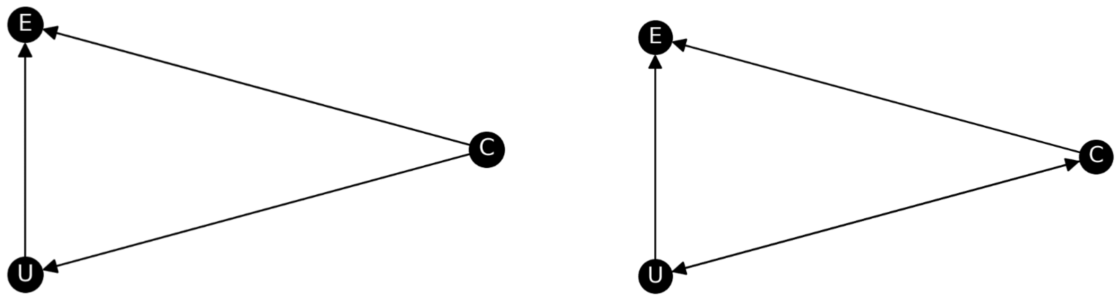
6.3 Demos and APIs

Reminder: Several friendly hooks here helping direct you back to the previous Section 6.2, the next Chapter 7, the current Chapter 6, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

6.3.1 Quick Tutorials

```python
from hybrid_algorithms import nonlinear_mlc
from utilities import scaling # Fine-tune the data generation by scaling.
import numpy as np
np.random.seed(42)

### Data Generation
# Simulate latent confounding to the data generation (uncomment the code if you want to test).
# pa_c = np.random.normal(size=2000) # Add an unobserved parent variable.
# c = np.random.normal(size=2000) + scaling(pa_c, 0.5)
# u = scaling(np.cos(c), 1) + scaling(np.random.normal(size=2000), 0.1) + scaling(pa_c, 0.5)

# Simulate the data generation with non-linear functional relations (without latent confounding).
c = np.random.normal(size=2000) # e.g. C is an exogenous variable
u = scaling(np.cos(c), 1) + scaling(np.random.normal(size=2000), 0.1) # C -> U
e = scaling(np.sin(c), 1) + np.sin(u) + scaling(np.random.normal(size=2000), 0.1) # C->E and U->E
dataset_a_without_confounding = np.array([c, e, u]).T

### Automation of Causal Inference
nonlinear_mlc.fit(dataset=dataset_a_without_confounding)

### Graph Visualization
array = nonlinear_mlc.adjacency_matrix_
draw_graph_from_ndarray(
    array=array,
    # Rename the graph nodes to consist with the data column.
    rename_nodes=['C', 'E', 'U']
)
plt.show()
```

Listing 6.4: *Quick Tutorials for Causal Discovery in Python.*



(**a**) *Causal Graph **Without** Latent Confounding* (**b**) *Causal Graph **With** Latent Confounding*

Figure 6.1: *Visualization of Nonlinear Causal Discovery Without and With Latent Confounders (e.g. Whether the nonlinear causal relationship between U and C is identifiable from observational (simulated) data).*

My Annotation: Three important points worth noting from the code sample in Listing 6.4:

- The data generation setting essentially reflects our prior Causal Assumption (I've introduced in Section 3.4 in this report). The assumption does not necessarily hold over the established dataset we want to analyze. Keeping this in mind might be helpful for us to objectively interpret the hypothetical causation learned from the empirical data.
- Insights into Figure 6.1b lie in the Partially Directed Acyclic Graphs (PDAGs) as the output (I've introduced in Section 4.1.1 in this report). That is because the bi-directed edge $U \leftrightarrow C$ that cannot be determined suggests the existence of latent confounder. Remarkably, the $C \rightarrow E$ (cause -> effect) is still identifiable, even if the indirected latent confounding $C \leftarrow \overline{pa}_C \rightarrow U \rightarrow E$ (raised by the unobserved parent $\overline{pa}_C$) persists between C and E (remind that this is the key finding discussed in our work, which has been introduced in Section 5.2).
- We fine-tune the scale of the "causal strength" among variables by the `scaling` function since theory-based causality-algorithms are always sensitive to the empirical data. This is excusable, as it is known that the causal assumption such as causal faithfulness (I've introduced in Section 3.3.5 in this report) is untestable (thus not-guaranteed) in practical.

6.3.2 APIs Overview and Close-up

The previous Section 6.3.1 showcases a single case of causal discovery through manual parameter assignment on simple structural data generation. Table 6.1 in this section further provides automatic utilities available to the task, with particularly an API close-up for the usage of causality-based data generation shown in Table 6.2.

Table 6.1: *APIs as to the Workflow in Causal Discovery.*

APIs Category	APIs Description	
	Reference/Class	**Calling Interface/Parameter Setup**
Hybrid-based Algorithms	NonlinearMLC (X. Chen et al., 2023b) MLCLiNGAM (W. Chen et al., 2021a)	`hybrid_algorithms.NonlinearMLC` `hybrid_algorithms.MLCLiNGAM`

SCMs Generator	CAMs (Bühlmann et al., 2014)
	ANMs (Hoyer et al., 2008)
	LiNGAM (Shimizu et al., 2011)
		`Generator.run_data_generation`
		`-- / setup="causal model"`
		`-- / setup="causal model"`

Causal Graph Evaluator	Structural Precision
	Structural Recall
	Structural F1 Score
		`Evaluator.precision_pairwise`
		`Evaluator.recall_pairwise`
		`Evaluator.f1_score_pairwise`

Table 6.2: *An API close-up for the Usage of the SCMs Generator in Table 6.1.*

Parameter Name	**Parameter Description**
`graph_node_num (int)`	Number of the vertex in a causal graph (ground-truth), which represents the number of the variable given a causal model (recommend: < 15)[required].
`sample (int)`	Size of the dataset generated from the SCMs (recommend: < 10000)[required].
`causal_model (str)`	**Structural-identifiable SCMs simulation in light of related literature, e.g. LiNGAM (str: lingam), CAMs/ANMs (str: hybrid_nonlinear).**
`noise_type (str)`	Structural-identifiable SCMs simulation in light of related literature. e.g. Gaussian (str: Gaussian), uniform distribution as non-Gaussian (str: non-Gaussian).
`sparsity (float)`	Control the sparsity of a causal graph (ground-truth) (recommend: 0.3).

7

Results, Discussion, and Related Work

This section recapitulates three of the implications related to my undergraduate research work (X. Chen et al., 2023b). Section 7.1 introduces a professional software that integrates implementation of the causality algorithms mentioned in Section 5.5 and Section 6.1. Section 7.2 introduces a simple application as to causal discovery over fMRI (functional Magnetic Resonance Imaging) data, in particular with delineation on how to evaluate performance of causal discovery. Finally, Section 7.4 introduces the outline of the temporal background for future work in causal discovery.

Reminder: Several friendly hooks here that direct you back to the previous Chapter 6, the next Chapter 8, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

7.1 CADIMULC: A Light Python Package for Hybrid Causal Discovery

CADIMULC is a Python package standing for: CAusal DIscovery with Multiple Latent Confounders. The address of online user interface and document with respect to the package is:
https://xuanzhichen.github.io/cadimulc/.

7.1.1 Technical Support

The package development partially relies on causal-learn (Zheng et al., 2024), an open-source python library for causal discovery. Independent modules such as (conditional) independence tests facilitates our custom needs in relation to the development of hybrid-based algorithmic framework.

7.1.2 Design Philosophy

CADIMULC aims to provide easy-to-use light APIs to learn an empirical causal graph from generally raw data with relatively efficiency. It integrates implementations of hybrid-based approaches involving the popular MLC-LiNGAM algorithm (W. Chen et al., 2021a), along with the ”micro” workflow of causal discovery, such as data generation, learning results evaluation, and graphs visualization.

Additional Notes: The MLC-LiNGAM approach (in IEEE-TNNLS, 2021) (W. Chen et al., 2021a) with its complete Python implementation are available in Appendix B.

7.1.3 Utilities, Demos, and APIs

To ensure coherence and completeness for Chapter 6 ”*Programming (Code Samples)*”, technical information related to the package is uniformly placed in Section 6.3.2 ”*APIs Overview and Close-up*”.

7.2 Inferring Causation Among Brain Regions over fMRI Data

Reminder: Several friendly hooks here helping direct you back to the previous Section 7.1, the next Section 7.3, the current Chapter 7, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

7.2.1 Neuroscience Background

The established fMRI dataset[1] in neuroscience is well-known for its mathematical basics of (nonlinear) dynamic causal models (Friston et al., 2003), with the nonlinearity setting catering to our research interest in this work. We then selected the fMRI-dataset (NetSim-3) that characterizes the temporal signals sampled from individuals (left panel in Figure 7.1), and that entails causal interactions among 15 distinct spatial regions (Regions of Interest, ROI) (right panel in Figure 7.1).

Hence, concerning the research topic in hidden confounding, the goal is to discover the causal structures (e.g. the mapping networks based on brain functions) over brain regions (denoted as the variable X_i) under the circumstances with omitted variables (shown red in Figure 7.1).

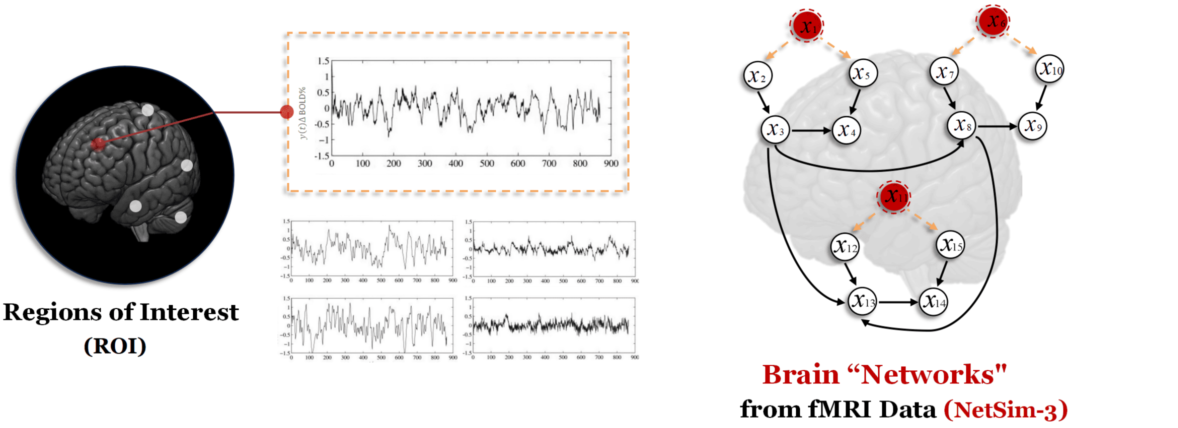


Figure 7.1: *Illustration of the causal structure (with latent confounders) with respect to ROI based on fMRI data. Omitted regions (namely the latent confounder) are marked as red color (Image reprinted from X. Chen et al., 2023b).*

7.2.2 Experimental Summary

Specifically, we first prioritized an increasing sequence of variables associating with brain regions (e.g. x_1, x_6, and x_{11}) as latent confounders by omitting them from original dataset. Then, given the fact that primary causal discovery methodologies discussed in this report are in the non-temporary category, we further reconstructed a non-temporal dataset via random sampling by giving a proper width-fixed time window. The dataset was processed by sampling with a size of 1000 from randomly selective 5 individuals, given width-fixed time windows with length of 200. Ultimately, we performed causal discovery approaches to the dataset.

7.2.3 Causal Discovery Baseline

We use the following causal discovery algorithms as the baseline methods: PC (Spirtes et al., 2000), FCI (Spirtes et al., 1991), RESIT (Peters et al., 2014), and CAM-UV (Maeda et al., 2021). PC is a constraint-based approach assuming causal sufficiency. Accordingly, FCI servers as an extension of PC algorithm, applying to causal inference with latent confounders. RESIT and CAM-UV are

[1] `https://www.fmrib.ox.ac.uk/datasets/netsim/index.html`

categorized to functional-based approaches. As a variant of DirectLiNGAM (Shimizu et al., 2011), RESIT assumes that non-linear additive models hold as for the data generation without presence of latent confounders. CAM-UV (Maeda et al., 2021), however, further assumes the existence of (general) unobserved variables, tending to avoid the incorrect causal inference. The usages of the baseline methods stated above refer to the python package causal-learn[2].

7.2.4 Causal Discovery Evaluation

We use precision, recall, and F1 score as the evaluation indicators for the estimated causal graphs reconstructed by different algorithms. Amidst the experiment, notice that we only extracted directed edges from the adjacency matrix or directly obtained causal pairs for calculating the indicators.

Before introducing typical metrics, **it is notable to outlines basic evaluation modules by analogizing classification metrics (in realms of machine learning) to causal structures discovery**: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). Common evaluation metrics defined around the comparison and trade-off between the estimated causal graph and the true causal graph (ground truth) will then be listed in the following.

- TP and FP can be seen as a measure of the causation identified in the estimated causal graph.
  * TP can be analogized to the total number of variable pairs (causal edges) identified in the estimated causal graph that are consistent with the causal relationships present in the true causal graph, quantifying the correctly estimated causal relationships between variables.
  * FP can be analogized to the total number of incorrectly identified variable pairs in the estimated causal graph that do not have causal relationships in the true causal graph.
- TN and FN are measures of the causation not identified in the estimated causal graph.
  * TN can be analogized to the total number of variable pairs in the estimated causal graph that are not identified but are consistent with the absence of causal relationships in the true causal graph, quantifying the independency correctly estimated between variables.
  * FN can be analogized to the total number of variable pairs (causal edges) that are missed in the estimated causal graph but have causal relationships in the true causal graph.

- **Precision** refers to the proportion of correctly identified causal relationships in the estimated causal graph among all estimated causal relationships. In other words, the higher the precision in the estimated causal graph, the more reliable the identification of causal relationships between variable pairs in the estimated causal graph.

$$Precision = \frac{TP}{TP + FP}. \tag{7.1}$$

- **Recall** refers to the proportion of correctly identified causal relationships in the estimated causal graph among all true causal relationships. Alternatively, the higher the recall in the estimated causal graph, the more it can encompass the causation over the true structure.

$$Recall = \frac{TP}{TP + FN}. \tag{7.2}$$

[2] `https://causal-learn.readthedocs.io/en/latest/`

- F_1 **Score** is the harmonic mean of precision and recall, integrating the advantages of both and serving as an overall measure of the effectiveness of causal discovery.

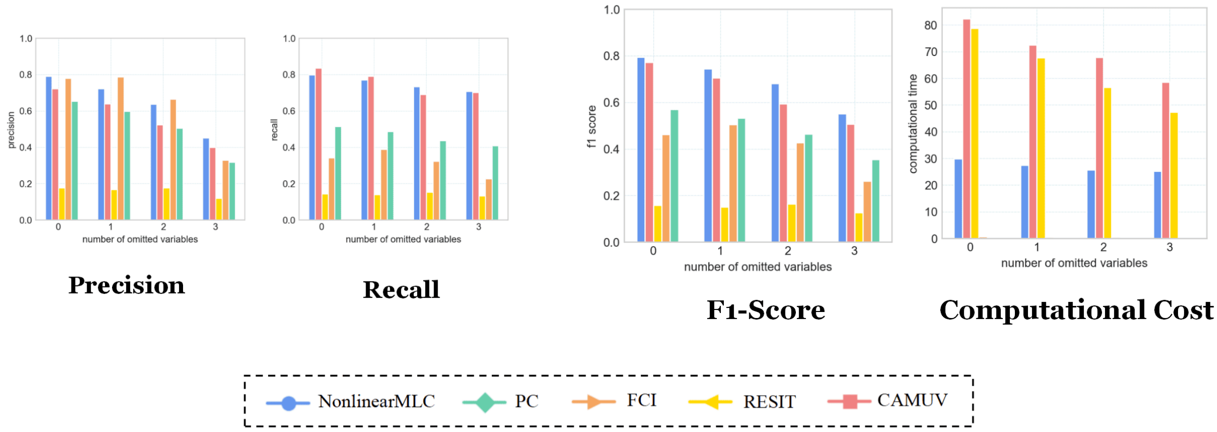
$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{7.3}$$

7.2.5 Experimental Finding

Notice that the relative simplicity of causal structures implied by fMRI data reduces the *Nonlinear-MLC* algorithm's advanced maximal-clique-based causal inference (described in Section 5.5 in this report) into the directed pairwise causal inference. Meanwhile:

- the CAM-UV algorithm (Maeda et al., 2021) might infer more hypothetical causal connections (including the redundant connections) without the foundation of the causal skeleton, which rendered its performance with marginally higher recalls but lower precision than ours.
- Despite of the well-perform precision by the FCI algorithm (Spirtes et al., 1991), it actually determined a small fraction of causal directions that contributes little to the recall.

Hence, **we conclude** that performance of the *Nonlinear-MLC* algorithm inclines to a slight advantage in the comprehensive F1 score with practicably lower computational time. The aforementioned analysis is illustrated in the following Figure 7.2.



Precision **Recall** **F1-Score** **Computational Cost**

Figure 7.2: *Performance evaluations in terms of precision, recall, and f1-score on fMRI-dataset (NetSim-3). (Image reprinted from X. Chen et al., 2023b).*

7.2.6 Additional Simulated Experiment

We also technically simulated the causal data generation to test our algorithm over synthetic dataset. Details referring to it can be found in my work (X. Chen et al., 2023b) (*Section 5.2 Performance on Simulated Causal Models (Non-linear MLC) Data*).

Briefly speaking, Figure 7.3 illustrates the average performance of *Nonlinear-MLC* compared with baseline methods. Except for the case of causal sufficiency — precision of our method is slightly lower than the CAM-UV algorithm — *Nonlinear-MLC* outperforms others in presence of latent confounders. Table 7.1 further demonstrates that our method is robust against the changes as to different sample sizes and dimensions.

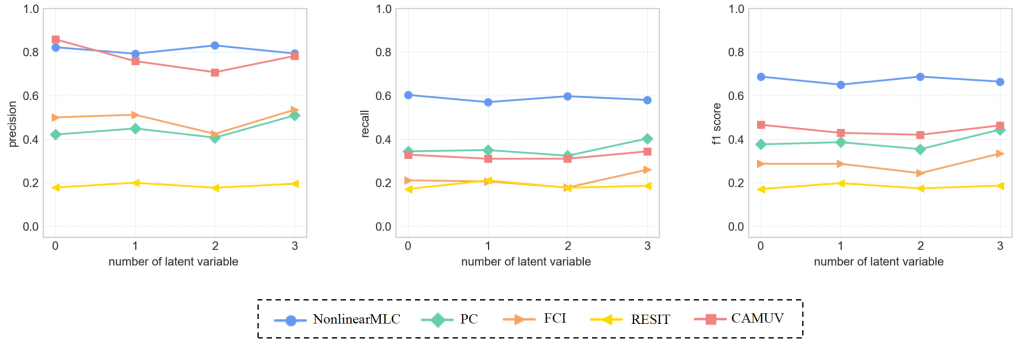


Figure 7.3: *Performance evaluations on simulated data with different numbers of latent confounders.*

Table 7.1: *Performance on Simulated Causal Models (NonlinearMLC). Sensitivity as to samples and dimensions, along with associating computational cost (Image reprinted from X. Chen et al., 2023b).*

Algorithm	F1 Score						Computational Time			
	Sample Size			Dimension			Number of Latent Confounder			
	500	1000	1500	5	10	15	0	1	2	3
PC	0.347	0.385	0.421	0.322	0.385	0.372	0.05	0.07	0.05	0.01
FCI	0.217	0.261	0.473	0.304	0.261	0.197	0.18	0.21	0.17	0.17
RESIT	0.152	0.169	0.174	0.277	0.169	0.124	29.78	29.73	29.82	29.86
CAMUV	0.265	0.374	0.586	0.625	0.374	0.419	14.7	17.59	15.14	15.16
NonlinearMLC	**0.623**	**0.661**	**0.735**	**0.851**	**0.661**	**0.627**	9.42	10.62	11.26	11.67

7.3 Question-Oriented Informal Discussion

Since the schedule of publishing this paper was eventually cancelled, the light discussion in this section were additionally listed. The following question-oriented discussion will specify supplemental perspectives as for this paper, by reviewing some of the equally important ideas (from my personal point of view) during my journey of finishing the work.

Reminder: Several friendly hooks here helping direct you back to the previous Section 7.2, the next Section 7.4, the current Chapter 7, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

7.3.1 Contribution on Causal Identification

1. Does the work in this paper truly tackle the issue of ”Multiple Latent Confounders”?

No quite, I have to admit. Initially I just wanted to ”extend” the repertoire of our previous work MLC-LiNGAM [W. Chen et al., 2021a], an causal discovery algorithm serves in a linear spectrum, by utilizing the conventional (non-linear) additive noise models (ANMs). To this end, I kept that abbreviation ”MLC” (Multiple Latent Confounders) for echoing the series of our work.

However, this might cause a slight exaggeration for the *Nonlinear-MLC* algorithm in this paper because I gradually found that non-linearity in causal inference is tricky than what I have imagined. The ”idea of extension” did not fully make sense due to the fact that the (linear) causal discovery strategies, which has paid off in MLC-LiNGAM, cannot just directly fit for *NonlinearMLC*. Technically speaking, in presence of multiple latent confounders, a LiNGAM (Linear Non-Gaussian Additive Model) would tend to hold after linear regression, whereas an ANM (non-linear Additive Noise Model) is distortion-prone via ”inadequate non-linear regression”. In essence, this therein results in the marjor difference between the MLC-LiNGAM and the *Nonlinear-MLC* algorithm.

Thus, it is partially the reason why I spent the time than anticipation on this paper. With the help from my advisor Wei Chen, fortunately, we alternatively discovered the *Latent-ANMs* lemma. Despite the lemma primarily functions as a ”fine-grained” theory, it might be simper and more distinctive in articulating the non-linear identification, which describes the relations between the cause-variable and the other unobserved patents of the effect-variable.

7.3.2 Limitation on Causal Algorithms

2. What is the limitation of the *Nonlinear-MLC* algorithm?

Though I have featured *Nonlinear-MLC* with emphasis on its theory-guided advantages, such as ”maximal clique patterns” and ”hybrid methodology”, I would like to say the meaning of the algorithm is more about the practicable causal discovery program on its own.

The strategy of maximal-clique-based causal inference, for instance, do strength the empirical performance of *Nonlinear-MLC*, whereas the algorithm in practice (according to my observation while developing the program) does not necessarily obey this ”fine-grained” theoretical strategies all the way (e.g. mostly a maximal clique includes the vertexes that are not more than 3, excluding the necessity for comprehensive analysis given such a simple structure). On top of that, restricted in a established hybrid-based framework, *Nonlinear-MLC* might sometimes become susceptible to the so-called cascading errors—an incorrect estimated causal skeleton (in the first stage of the algorithm) can compromise the subsequent non-linear regression and independence tests.

Bottomline, I think the ideas of *Nonlinear-MLC*, good or bad, would largely depend on the feedback from users in different fields who want to give the non-linear causal inference a shot. By applying *Nonlinear-MLC*, I wish the users more or less are able to get a rough understanding about the causation with respect to the field-related data they are interested in.

7.3.3 Future Work

3. Would the algorithm be extended to apply for time series data in the future work?

No, and I would not recommend that my follow-up work is done to serve as a ”time-series version” of *Nonlinear-MLC*, though it might be a good way to quickly grasp the idea and yield another paper for utilitarian purpose. Temporal causal discovery has recently been a popular topic, but I wish we could dig deeper instead of directly launching a parallel extension. However, we did have a work (X. Chen et al., 2023a) partially investigating the latest progress with respect to time-series causal discovery, which I will briefly introduce in the next section.

7.4 Relevant Work: Introduction to Temporal Causal Discovery

Reminder: Several friendly hooks here helping direct you back to the previous Section 7.3, the next Chapter 8, the current Chapter 6, the table of Contents, or the Reading Guidance of This Report for Different Audiences.

In recent years, high-volume and high-dimensional data have been continuously developing. Time series data, as an ordered sequence of real values collected over time, often carries real-world information. Over the past few decades, numerous time series analysis methods based on various tasks such as classification (Ismail Fawaz et al., 2019; Lines et al., 2014), clustering (Aghabozorgi et al., 2015; Li et al., 2011), and forecasting (Wang et al., 2019; Weigend, 2018) have emerged. Among these, Temporal Causal Discovery (TCD) based on observational data — identifying causal-and-effect between different time series without relying on intervention — is also a notable task.

When it comes to the context of time series, in particular considering the autocorrelation among variables, nevertheless, causal graphs are typically aggregated or expanded along the time axis based on time windows, which further categorizing them into: Summary Causal Graphs, Window Causal Graphs, and Full-Time Causal Graphs. Different types of temporal causal graphs serve as the objectives for TCD in numerous systems of natural sciences or human society, such as climate science (Stips et al., 2016), efficacy assessment (Bica et al., 2020), and economic markets (Hiemstra et al., 1994), adding complexities as to dynamic systems analysis.

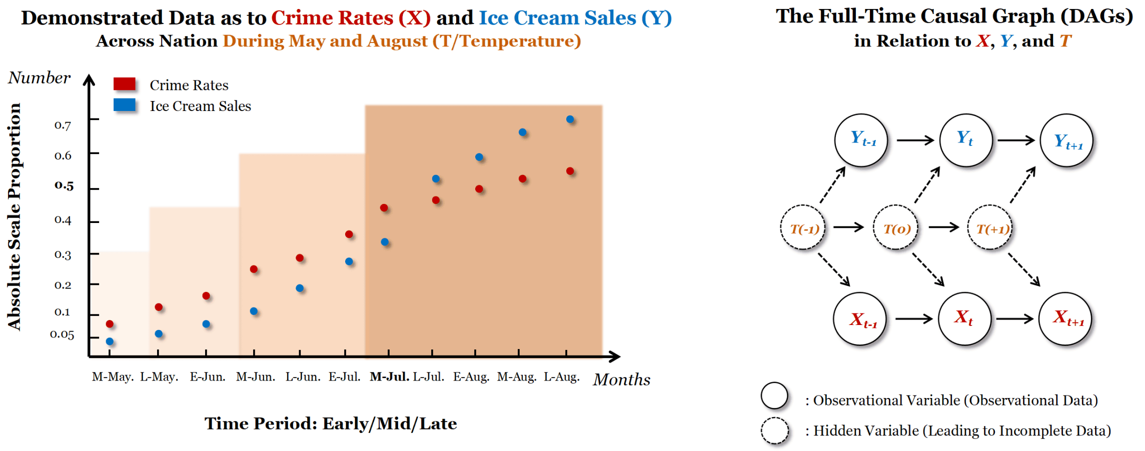


Figure 7.4: *A Time-Series-Based System Lacking Consideration of Causality (Thought Experiments) (Image reprinted from X. Chen et al., 2023a).*

On the other hand, time-series-based systems that lack consideration of causality may mislead people to draw incorrect or even absurd conclusions. **For example (Figure 7.4), the sales of ice cream in urban areas during the summer and the rise in local crime rates are clearly two time series variables that intuitively seem unrelated (non-causal), yet they may exhibit a high statistical correlation.** In fact, merely considering the time series of ice cream sales and crime rates as observational data is INCOMPLETE. A reasonable explanation is that the unobserved data related to urban temperatures (latent confounder) leads to this biased result; baking heat during urban hot summers simultaneously stimulate increases in both ice cream sales and potential crime rates.

So far, several reviews (Moraffah et al., 2021; Assaad et al., 2022; Gong et al., 2023; Hasan et al., 2023) have comprehensively summarized the progress in TCD based on observational data from uniquely different perspectives, including causal-effect analysis (Moraffah et al., 2021), practical applications of various causality methods (Assaad et al., 2022), data type of event-sequence(Hasan et al., 2023), and the integration of non-temporal and temporal causal discovery issues (Hasan et al., 2023). However, topics regarding TCD based on incomplete data — observational data with hidden variables or missing samples — are still insufficient. Therefore, our work (X. Chen et al., 2023a) provides a survey that is meant to investigate the latest research progress in relation to this kind of topic.

8

My Humble and Trivial Opinions to Causality

Were I to summarize my experience, reflection, and even audacity regarding causality and AI — or Causal Discovery in particular — the first sentence comes to my mind is: The quintessence of the theoretically inferred causation rests metaphorically on our ignorance against God or Nature.

By ignorance I mean that even a prophetic model may need not be anticipated to carry omniscience. A Bayesian model characterizes uncertainty, virtuous in being capable of unrestrictedly predicting any relationship through an almighty inverse formula complying with calculation logic. A causal model, by contrast, characterizes certainty, meticulous in specifying only the irreversible ”listening-to” relationship while encapsulating any other unspecificness into a humble term that represents disturbance or noise from ”God” or ”Nature”. Admittedly, physics and algebra have told us that nearly every science in formulae demands a backup from the equality sign — meaning, the so-called ”listening to” asymmetry is tenuous. Yet there is a light in the tunnel. It's not longer easy for one to swap terms on both sides of the equality sign, given an unspecific term that intrudes the purity and balance of the equation. If one struggles to inversely fit such unspecificness via untangling its composition, then in a metaphorical sense, he or she may be viewed as boastly demanding the omniscience that only ”God” or ”Nature” possesses.

A better vantage point, regarding the aforementioned conceptual causation, lies in the context of probability, a language that characterizes uncertainty. Since our ignorance within a causal model implies the model's tolerance for marginally uncertain accidents, all authoritative Causal Discovery methodologies operate resembling a behavior of unveiling the distribution regarding such ”accidents” out of the original data distribution, and studies their probabilistic property such as independency — akin to the by-product of causation (e.g. random accidents may imply independency).

Causal Discovery results in causal graphs as a blue print representing the causal model. Must such a blue print, however, to be in form of a 2-dimensional diagram that one can simply draw on a white sheet of paper, as what I have been taught in Causal Discovery? I really don't know. I may boldly envision the blue print present even at an abstract level with an infinite scale, as long as from which it makes it effortless (less dependent upon data) for AI to retrieve a kind of deterministic relationship. By effortlessness I also imply a potential computational process completed within a blink of eyes, probably propelled by an established parametric model on top of the blue print. We presume the blue print exists as a simple ”structural” avatar relative to that hidden parametric model. This ”structural” nature navigates AI through how to unveil prior parameters bottom-up through the structure, how to tweak the model causally, and how to convey posterior parameters top-down.

Finally, with a little bit digression, I admit that, at least for me, it is Judea Pearl's personal charisma that by itself adds a kind of eternally appealing sense into my understanding on causality. Pearl left me with a lasting impression as someone who is not only a Turing Award winner and an inventor of Bayesian Networks and Causal Models, but a miraculous survivor during his military service, a father who lost his son and who bore witness to human intolerance, as well as a life-time pursuer for human-level AI but has found himself helpless nowadays in contributing to this argument. My poor knowledge about causality thus become convoluted, perceptual, and even kind of being philosophized at the end, with the person behind it. Therefore, I couldn't find any other way to wrap up my emotional personal thoughts in this Chapter — also in the end of this report — better than paraphrasing his resonating advice to youngsters and his hope towards his legacy left for the next decade:

Words by Judea Pearl (***he's 88 years old in 2024***):

Ask yourself questions and solve them in your way, as opposed to merely accepting "NO" for an answer. ***Questions coming out of your brain are never dumb — Follow them, and try to understand them in your own way****. For example, there is a lot of inertia in the academia that is slowing down science. Dare to "against" your professor. I wrote the book of why (*Pearl et al., 2018*) in order to democratize common sense, in order to instill rebellious spirits in students, so they wouldn't wait until the professor gets everything down.*

In terms of me, I already have a tombstone carved: The Fundamental Law of Counterfactuals. It's a simple equation, putting causal counterfactuals in terms of a surgery on causal models — because everything follows from there. ***If you get that, all the rest follow****. I can also die in peace, and my students can derive all my life-time knowledge by mathematical means.*

Bibliography

Aghabozorgi, Saeed, Ali Seyed Shirkhorshidi, and Teh Ying Wah (2015). “Time-series clustering–a decade review”. In: *Information systems* 53, pp. 16–38.

Assaad, Charles K, Emilie Devijver, and Eric Gaussier (2022). “Survey and evaluation of causal discovery methods for time series”. In: *Journal of Artificial Intelligence Research* 73, pp. 767–819.

Bica, Ioana, Ahmed Alaa, and Mihaela Van Der Schaar (2020). “Time series deconfounder: Estimating treatment effects over time in the presence of hidden confounders”. In: *International Conference on Machine Learning*. PMLR, pp. 884–895.

Bühlmann, Peter, Jonas Peters, and Jan Ernest (2014). “CAM: Causal additive models, high-dimensional order search and penalized regression”. In: *The Annals of Statistics* 42.6, pp. 2526–2556.

Cai, Ruichu, Zhenjie Zhang, and Zhifeng Hao (2013). “Sada: A general framework to support robust causation discovery”. In: *International conference on machine learning*. PMLR, pp. 208–216.

Cai, Ruichu et al. (2019). “Triad constraints for learning causal structure of latent variables”. In: *Advances in neural information processing systems* 32.

Chen, Wei et al. (2021a). “Causal discovery in linear non-gaussian acyclic model with multiple latent confounders”. In: *IEEE Transactions on Neural Networks and Learning Systems*.

Chen, Wei et al. (2021b). “Fritl: A hybrid method for causal discovery in the presence of latent confounders”. In: *arXiv preprint arXiv:2103.14238*.

Chen, Xuanzhi (2024). “A Primer on Causal Diagram Learning”. In.

Chen, Xuanzhi, Wei Chen, and Ruichu Cai (2023a). “A Survey on Causal Discovery with Incomplete Time-Series Data”. In.

Chen, Xuanzhi, Wei Chen, and Ruichu Cai (2023b). “Non-linear Causal Discovery for Additive Noise Model with Multiple Latent Confounders”. In.

Friston, Karl J, Lee Harrison, and Will Penny (2003). “Dynamic causal modelling”. In: *Neuroimage* 19.4, pp. 1273–1302.

Gong, Chang et al. (2023). “Causal Discovery from Temporal Data: An Overview and New Perspectives”. In: *arXiv preprint arXiv:2303.10112*.

Gretton, Arthur et al. (2005). “Measuring statistical dependence with Hilbert-Schmidt norms”. In: *International conference on algorithmic learning theory*. Springer, pp. 63–77.

Hariton, Eduardo and Joseph J Locascio (2018). “Randomised controlled trials—the gold standard for effectiveness research”. In: *BJOG: an international journal of obstetrics and gynaecology* 125.13, p. 1716.

Hasan, Uzma, Emam Hossain, and Md Osman Gani (2023). “A Survey on Causal Discovery Methods for Temporal and Non-Temporal Data”. In: *arXiv preprint arXiv:2303.15027*.

Hiemstra, Craig and Jonathan D Jones (1994). “Testing for linear and nonlinear Granger causality in the stock price-volume relation”. In: *The Journal of Finance* 49.5, pp. 1639–1664.

Hoyer, Patrik et al. (2008). "Nonlinear causal discovery with additive noise models". In: *Advances in neural information processing systems* 21.

Ismail Fawaz, Hassan et al. (2019). "Deep learning for time series classification: a review". In: *Data mining and knowledge discovery* 33.4, pp. 917–963.

Koller, Daphane (2009). *Probabilistic Graphical Models: Principles and Techniques*.

Li, Lei and B Aditya Prakash (2011). "Time series clustering: Complex is simpler!" In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 185–192.

Lines, Jason and Anthony Bagnall (2014). "Ensembles of elastic distance measures for time series classification". In: *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, pp. 524–532.

Maeda, Takashi Nicholas and Shohei Shimizu (2021). "Causal additive models with unobserved variables". In: *Uncertainty in Artificial Intelligence*. PMLR, pp. 97–106.

Maeda, Takashi Nicholas and Shohei Shimizu (2024). "Use of prior knowledge to discover causal additive models with unobserved variables and its application to time series data". In: *Behaviormetrika*, pp. 1–19.

Moraffah, Raha et al. (2021). "Causal inference for time series analysis: Problems, methods and evaluation". In: *Knowledge and Information Systems* 63, pp. 3041–3085.

Neal, Brady (2020). "Introduction to causal inference". In: *Course Lecture Notes (draft)*.

Pearl, Judea (2009). *Causality*. Cambridge university press.

Pearl, Judea and Dana Mackenzie (2018). *The book of why: the new science of cause and effect.* Basic books.

Peters, Jonas, Dominik Janzing, and Bernhard Schölkopf (2017). *Elements of causal inference: foundations and learning algorithms*. The MIT Press.

Peters, Jonas et al. (2014). "Causal discovery with continuous additive noise models". In.

Qiao, Jie et al. (2021). "Causal discovery with confounding cascade nonlinear additive noise models". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 12.6, pp. 1–28.

Schölkopf, Bernhard (2022). "Causality for machine learning". In: *Probabilistic and causal inference: The works of Judea Pearl*, pp. 765–804.

Shimizu, Shohei et al. (2006). "A linear non-Gaussian acyclic model for causal discovery." In: *Journal of Machine Learning Research* 7.10.

Shimizu, Shohei et al. (2011). "DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model". In: *Journal of Machine Learning Research-JMLR* 12.Apr, pp. 1225–1248.

Spirtes, Peter and Clark Glymour (1991). "An algorithm for fast recovery of sparse causal graphs". In: *Social science computer review* 9.1, pp. 62–72.

Spirtes, Peter et al. (2000). *Causation, prediction, and search*. MIT press.

Stephenson, Todd Andrew (2000). *An introduction to Bayesian network theory and usage.*

Stips, Adolf et al. (2016). "On the causal structure between CO2 and global temperature". In: *Scientific reports* 6.1, p. 21691.

Tashiro, Tatsuya et al. (2014). "ParceLiNGAM: A causal ordering method robust against latent confounders". In: *Neural computation* 26.1, pp. 57–83.

Wang, Yuyang et al. (2019). "Deep factors for forecasting". In: *International conference on machine learning*. PMLR, pp. 6607–6617.

Weigend, Andreas S (2018). *Time series prediction: forecasting the future and understanding the past*. Routledge.

Wood, Simon N (2004). “Stable and efficient multiple smoothing parameter estimation for generalized additive models”. In: *Journal of the American Statistical Association* 99.467, pp. 673–686.

Zhang, Kun et al. (2012). “Kernel-based conditional independence test and application in causal discovery”. In: *arXiv preprint arXiv:1202.3775*.

Zheng, Yujia et al. (2024). “Causal-learn: Causal discovery in python”. In: *Journal of Machine Learning Research* 25.60, pp. 1–8.

Appendices

A

Appendix A

A.1 Proof of Lemma 1

We proved Lemma 1 by transferring variable descriptions beforehand — from the intuitive pairwise cause-and-effect $C \rightarrow E$ into the standard structure causal models (SCMs) with the variables $\boldsymbol{X} = \{x_1, x_2, \ldots, x_d\}$. The corresponding Lemma relative to SCMs is stated as the following.

Lemma 3 (***SCMs***)**:** *Assuming data generation procedures are consistent with Equation (5.3), the pairwise causal dependence between the effect-variable x_i and one of the associating cause-variables $x_j \in \boldsymbol{pa}_i$ is identifiable if and only if*

$$\{\xi_i \perp\!\!\!\perp x_j\} \wedge \{\xi_i := \varepsilon_i + \sum_{\ell_k \in \overline{\boldsymbol{pa}}_i} f_{ik}(\ell_k)\} \tag{A.1}$$

is satisfied, where ξ_i is denoted as an extensive noise (e.g. compared to the original noise ε_i that has satisfied $\varepsilon_i \perp\!\!\!\perp x_j$). The extensive noise ξ_i further models the multiple latent confounding from the multiple unobserved parents $\overline{\boldsymbol{pa}}_i$.

Taking the potential latent confounders $\ell_k \in \overline{\boldsymbol{pa}}_i$ into consideration, Lemma 1 (SCMs) provides an independent condition to identify the unambiguous causal directions $\{x_j \rightarrow x_i \mid x_j \in \boldsymbol{pa}_i\}$. Next, suppose we use $x_j \rightarrow x_i$ $(x_j = \boldsymbol{pa}_i)$ to represent any of the identifiable pairs satisfying Lemma 1 (SCMs).

Notice that the proof of Lemma 1 (SCMs) is equal to prove that the *Nonlinear-MLC* causal model only holds in the causal direction $x_j \rightarrow x_i$. According to Equation (5.3), we further formalize the generation procedure as to a correct causal model $\mathcal{M}_1$:

$$\mathcal{M}_1 : x_i := f_{ij}(x_j) + \mathcal{F}_i(\boldsymbol{\ell}_i) + \varepsilon_i. \tag{A.2}$$

Where $\mathcal{F}_i(\boldsymbol{\ell}_i) = \sum_{\ell_k \in \overline{\boldsymbol{pa}}_i} f_{ik}(\ell_k)$. Without loss of generality, we slightly distinguish the reversed non-linear function and the latent noise, in the sense that an inversed (incorrect) causal model $\mathcal{M}_2$ satisfies

$$\mathcal{M}_2 : x_j := \tilde{f}_{ji}(x_i) + \tilde{\mathcal{F}}_j(\boldsymbol{\ell}_j) + \tilde{\varepsilon}_j. \tag{A.3}$$

We factorize the marginal distribution (with multiple unobserved parents) entailed by both models:

$$p(x_i,\ x_j) = \sum_{\boldsymbol{\ell}} p(x_i,\ x_j \mid \ell)\, p(\ell) = \begin{cases} \sum_{\boldsymbol{\ell}_i}\ p(x_i \mid x_j, \ell, \mathcal{M}_1)\, p(x_j \mid \ell, \mathcal{M}_1)\, p(\ell \mid \mathcal{M}_1), \\ \sum_{\boldsymbol{\ell}_j}\ p(x_j \mid x_i, \ell, \mathcal{M}_2)\, p(x_i \mid \ell, \mathcal{M}_2)\, p(\ell \mid \mathcal{M}_2). \end{cases} \tag{A.4}$$

Notice that the independent noise ε is generalized into (the possibly dependence) ξ, along with the independence $\xi_i \perp\!\!\!\perp x_j$ entailed by the identifiable causal model $\mathcal{M}_1$:

$$\xi = \mathcal{F}(\boldsymbol{\ell}) + \varepsilon = \sum_{\ell_k \in \overline{\boldsymbol{pa}}} f_k(\ell_k) + \varepsilon, \quad \xi_i \perp\!\!\!\perp x_j. \tag{A.5}$$

Given likelihood functions $\mathcal{L} = \log p(\cdot)$ and injective relations between ξ_i and x_j ($\tilde{\varepsilon}_j$ and x_i), combining Equations (A.4) and (A.5) yields

$$\mathcal{L}(\mathcal{M}) = \begin{cases} \mathcal{L}_{\xi_i}(x_i - f_{ij}(x_j)) +\ \mathcal{L}_{x_j}(x_j), & \mathcal{M} = \mathcal{M}_1, \\ \mathcal{L}_{\tilde{\varepsilon}_j}\left(x_j - \tilde{f}_{ji}(x_i) - \tilde{\mathcal{F}}_j(\boldsymbol{\ell})\right) +\ \mathcal{L}_{x_i}(x_i), & \mathcal{M} = \mathcal{M}_2. \end{cases} \tag{A.6}$$

Additionally, we herein **emphasize** that the strict independence $\xi_i \perp\!\!\!\perp x_j$ ensures the expression of $\mathcal{L}(\mathcal{M} = \mathcal{M}_1)$ in Equation (A.6). In other words, the conditional independence (between ξ_i and x_j) is insufficient to yield that expression in form of regression-based replacement (e.g. replace $\mathcal{L}_{x_i|x_j,\boldsymbol{\ell}_i}(x_i)$ in eq.(A.4) by $\mathcal{L}_{\xi_i}(x_i - f_{ij}(x_j))$ in eq.(A.6)). **The reason** is given by the non-linearity, which implies that the variables' non-linear interaction, compared with linearity, will compromise the effect of regression (**recall** the Introduction and Section 3 in the paper).

Based on the formalism shown in Equation (A.6)), we continue the rest of the proof framework by following the **ANMs identification** [Hoyer et al., 2008]. Assuming $\tilde{f}$ is third order differentiable we obtain

$$\frac{\partial}{\partial x_j}\left(\frac{\partial^2 \mathcal{L}(\mathcal{M})/\partial x_j^2}{\partial^2 \mathcal{L}(\mathcal{M})/\partial x_i \partial x_j}\right) = 0, \quad \mathcal{M} = \mathcal{M}_2. \tag{A.7}$$

Notice that this is not hold when $\mathcal{M} = \mathcal{M}_1$. To see this, imply

$$\frac{\partial^2 \mathcal{L}(\mathcal{M}_1)}{\partial x_j\, \partial x_i} = -f'_{ij}\mathcal{L}''_{x_i}, \tag{A.8}$$

and

$$\frac{\partial^2 \mathcal{L}(\mathcal{M}_1)}{\partial x_j^2} = \mathcal{L}''_{\xi_i}(f'_{ij})^2 - \mathcal{L}' f''_{ij} + \mathcal{L}''_{x_j}, \tag{A.9}$$

then we obtain the analogical differential equation (compared with Equation (A.7)) constructed by $\mathcal{M}_1$:

$$\frac{\partial}{\partial x_j}\left(\frac{\frac{\partial^2 \mathcal{L}(\mathcal{M}_1)}{\partial x_j^2}}{\frac{\partial^2 \mathcal{L}(\mathcal{M}_1)}{\partial x_i \partial x_j}}\right) = -2f_{ij}'' + \frac{\mathcal{L}_{\xi_i}' f_{ij}''' - \mathcal{L}_{x_j}'''}{\mathcal{L}_{\xi_i}'' f_{ij}'} + \frac{\mathcal{L}_{x_j}'' f_{ij}'' - \mathcal{L}_{\xi_i}' (f_{ij}'')^2}{\mathcal{L}_{\xi_i}'' (f_{ij}')^2} + \frac{\mathcal{L}_{\xi_i}' \mathcal{L}_{\xi_i}''' f_{ij}'' - \mathcal{L}_{x_j}'' \mathcal{L}_{\xi_i}'''}{(\mathcal{L}_{\xi_i}'')^2}. \tag{A.10}$$

Notice that here we omit the variable inside the function notation.

In order to vanish Equation (A.10) (if both of the forward causal model $\mathcal{M}_1$ and backward causal model $\mathcal{M}_2$ hold over the joint probability $p(x_i,\ x_j)$), we are supposed to obtain the following (linear inhomogeneous) differential equation [Hoyer et al., 2008] for every fix x_i given $\mathcal{L}_{\xi_i}'' \cdot f_{ij}' \neq 0$. It is given by

$$\mathcal{L}_{x_j}(x_j)''' = \mathcal{L}_{x_j}(x_j)'' \phi(x_j,\ x_i) \ + \ \eta(x_j,\ x_i)\,, \tag{A.11}$$

where $\phi(x_j,\ x_i)$ and $\eta(x_j,\ x_i)$ are defined by

$$\phi(x_j,\ x_i) = -\frac{\mathcal{L}_{\xi_i}''' f_{ij}'}{\mathcal{L}_{\xi_i}''} + \frac{f_{ij}''}{f_{ij}'}, \tag{A.12}$$

and

$$\eta(x_j,\ x_i) = -2\mathcal{L}_{\xi_i}'' f_{ij}'' f_{ij}' + \mathcal{L}_{\xi_i}' f_{ij}''' + \frac{\mathcal{L}_{\xi_i}' \mathcal{L}_{\xi_i}''' f_{ij}'' f_{ij}'}{\mathcal{L}_{\xi_i}''} - \frac{\mathcal{L}_{\xi_i}' (f_{ij}'')^2}{f_{ij}'}. \tag{A.13}$$

Therefore, from Equation (A.11) - (A.13) we conclude that the hypothetical $\mathcal{L}_{x_j}$ admitting a backward causal model is limited in a three-dimensional, which contradicts our priority that all possible $\mathcal{L}_{x_j}$ should be infinite-dimensional [Hoyer et al., 2008]. That is, from the perspective of generic, the *Nonlinear-MLC* causal model only holds in $x_j \longrightarrow x_i$ and can not be inverted.

A.2 Proof of Corollary 1

Likewise, Corollary 1 was proven provided the context of standard structure causal models (SCMs). The associating Corollary with respect to SCMs is claimed as the following.

Corollary 2 (***SCMs***)**:** *Assuming data generation procedures are consistent with Equation (5.3), the pairwise causal dependence between the effect-variable x_i and one of the associating cause-variables $x_j \in \boldsymbol{\mathcal{M}}_{ij}$ is identifiable if and only if*

$$\{x_i - \mathcal{R}_i(\boldsymbol{\mathcal{M}}_{ij}^* \ \cup \ \hat{\boldsymbol{pa}}_i)\} \perp\!\!\!\perp \ x_j \tag{A.14}$$

is satisfied, where $\mathcal{R}(\cdot)$ denotes the non-linear regressor, $\boldsymbol{\mathcal{M}}_{ij}^* := \boldsymbol{\mathcal{M}}_{ij} \backslash \ \{x_i\}$, and $\hat{\boldsymbol{pa}}_i \subseteq \boldsymbol{pa}_i$. In the view of computing memory in (constraint-based) algorithms, $\hat{\boldsymbol{pa}}$ denotes the determined *parent relations, whereas $\boldsymbol{\mathcal{M}}_{ij}$ represent the variables (including x_j and x_i) whose relations remain undetermined within the possible maximal cliques.*

Providing identifiable causal directions $\{x_j \longrightarrow x_i \mid x_j \in \boldsymbol{\mathcal{M}}_{ij}\}$, we assume the causal direction as $x_j \longrightarrow x_i$ to represent any of the identifiable pairs satisfying Corollary 1 (SCMs). The data generation process of the variable x_i can be formulated as

$$x_i := f_{ij}(x_j) + \sum_{x_t \in \boldsymbol{pa}_i \setminus \{x_j\}} f_{it}(x_t) + \sum_{\ell_k \in \overline{\boldsymbol{pa}}_i} f_{ik}(\ell_k) + \varepsilon_i, \tag{A.15}$$

According to the causal additive models (CAMs) [Bühlmann et al., 2014], the empirical (non-linear) regressor $\mathcal{R}_i$ (for the explaining variable x_i) of general additive models (GAMs) [Maeda et al., 2021] is defined by

$$\mathcal{R}_i := g_{ij}(x_j) + \sum_{x_t \in \hat{\boldsymbol{pa}}_i} g_{it}(x_t) + \sum_{x_r \in \boldsymbol{\mathcal{M}}^*_{ij}} g_{ir}(x_r), \tag{A.16}$$

where $g(\cdot)$ denotes the empirical regression function selected from GAMs.

Since $\mathcal{R}_i$ is decomposed into several specific parts to cancel the effect of hypothetical cause-variables, we substitute the regressor $\mathcal{R}(\cdot)$ in Corollary 1 (SCMs) with Equation (A.15) and (A.16). We conclude

$$H_i(x) \perp\!\!\!\perp x_j, \tag{A.17}$$

where $H_i(x)$ is defined by

$$H_i(x) := \big\{f_{ij}(x_j) - g_{ij}(x_j)\big\} + \left\{ \sum_{x_t \in \boldsymbol{pa}_i \setminus \{x_j\}} f_{it}(x_t) - \{ \sum_{x_t \in \hat{\boldsymbol{pa}}_i} g_{it}(x_t) + \sum_{x_r \in \boldsymbol{\mathcal{M}}^*_{ij}} \hat{g_{ir}}(x_r)\} \right\} + \{ \sum_{\ell \in \overline{\boldsymbol{pa}}_i} f_{ik}(\ell_k) + \varepsilon_j\}, \tag{A.18}$$

We **highlight** that the variable set (including x_i) consisting of a maximal clique $\mathcal{M}_{ij}$ might involve the correct (but undetermined) parent relations in the view of algorithmic memory:

$$\exists\, x_r \in \boldsymbol{\mathcal{M}}^*_{ij},\ x_r \not\perp\!\!\!\perp x_i \;\Rightarrow\; x_r \;\in \boldsymbol{pa}_i. \tag{A.19}$$

In light of Lemma 1, the anticipant independence (**recall** Section 4.1 in the paper) is defined as

$$x_j \;\perp\!\!\!\perp \boldsymbol{pa}_i \setminus \{\hat{\boldsymbol{pa}}_i \,\cup\, \boldsymbol{\mathcal{M}}^*_{ij}\}. \tag{A.20}$$

We then consider three of the independence combinations of $H_i(x)$ relative to Equation (A.17). We have

(1) $f_{ij}(x_j) - g_{ij}(x_j) = 0$, which is ideally required by the GAMs regression.

(2) $Z_i(x) \perp\!\!\!\perp x_j$, where $Z_i(x)$ is defined by

$$Z_i(x) := \sum_{x_t \in \boldsymbol{pa}_i \setminus \{x_j\}} f_{it}(x_t) - \{ \sum_{x_t \in \hat{\boldsymbol{pa}}_i} g_{it}(x_t) + \sum_{x_r \in \boldsymbol{\mathcal{M}}^*_{ij}} \hat{g_{ir}}(x_r)\}. \tag{A.21}$$

Notice that assuming $\{x_j \perp\!\!\!\perp \boldsymbol{pa}_i \setminus \{\hat{\boldsymbol{pa}}_i \cup \boldsymbol{\mathcal{M}}^*_{ij}\}\}$ by Equation (A.20) enforces Equation (A.21) to vanish into irrelevant regressing residuals with respect to x_j.

(3) $\xi_i \perp\!\!\!\perp x_j$, where ξ_i is the extensive noise (in the data generation procedure, Equation (1)) defined by

$$\xi_i := \sum_{\ell \in \overline{\boldsymbol{pa}}_i} f_{ik}(\ell_k) + \varepsilon_j. \tag{A.22}$$

The independence for the identifiable $x_j \rightarrow x_i$ has already required by Lemma 1 (SCMs).

Thus, the independence implied by Equation (A.17) eventually reduces to

$$\{Z_i(x) \cup \xi_i\} \perp\!\!\!\perp x_j, \; H_i(x) := 0 + Z_i(x) + \xi_i, \tag{A.23}$$

which represents Corollary 1 (SCMs) and is further satisfied by the sub-conditions (1)-(3).

B

Appendix B

B.1 Python Implementation of the MLC-LiNGAM Algorithm

```python
# Author:  Xuanzhi CHEN <xuanzhichen.42@gmail.com>
# License: MIT License

from __future__ import annotations

# hybrid causal discovery framework
from .hybrid_framework import HybridFrameworkBase

# auxiliary modules in causality instruments
from cadimulc.utils.causality_instruments import (
    get_residuals_scm,
    conduct_ind_test,
)
# linear regression
from sklearn.linear_model import LinearRegression

# basic
from cadimulc.utils.extensive_modules import (
    check_1dim_array,
    copy_and_rename
)
from numpy import ndarray

import numpy as np
import networkx as nx
import copy as cp
import time
import warnings
warnings.filterwarnings("ignore")

class GraphPatternManager(object):
    # Implementation

class MLCLiNGAM(HybridFrameworkBase):
    """
    *MLC-LiNGAM stands* for a **hybrid** causal discovery method for the **LiNGAM** approach
    with **multiple latent confounders**.
    It serves as an enhancement of **LiNGAM<sup>*</sup>** via combining the advantages of
    **constraint-based** and **functional-based** causality methodology.

    !!! note "The LiNGAM causal discovery approach"
        LiNGAM, the linear non-Gaussian acyclic model, is known as one of the
        [structural-identifiable SCMs](https://xuanzhichen.github.io/cadimulc/generation/).

    ***MLC-LiNGAM* was proposed to alleviate the following issues**:

    - how to detect the latent confounders;
    - how to uncover the causal relations among observed and latent variables.

    <!--
    References:
    Chen, Wei, Ruichu Cai, Kun Zhang, and Zhifeng Hao.
    "Causal discovery in linear non-gaussian acyclic model with multiple latent confounders. "
    *IEEE Transactions on Neural Networks and Learning Systems.* 2021.

    Shimizu, Shohei, Patrik O. Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan.
    A linear non-Gaussian acyclic model for causal discovery.
    *Journal of Machine Learning Research.* 2006.
    -->
    """

    def __init__(
            self,
            pc_alpha: float = 0.05,
            _latent_confounder_detection: list[list] = []
    ):
        """
        Parameters:
            pc_alpha:
                Significance level of independence tests (p_value), which is required by
                the constraint-based methodology incorporated in the initial stage of
                the hybrid causal discovery framework.

        <!--
        Attributes:
            _latent_confounder_detection:
                The list elements given by `_latent_confounder_detection` are
                undirected maximal cliques after stage-III learning, suggesting that the
                variables within the undirected maximal clique share an unknown common cause.
        -->
        """

        HybridFrameworkBase.__init__(self, pc_alpha=pc_alpha)
        self._latent_confounder_detection = _latent_confounder_detection

    # ### AUXILIARY COMPONENT(S) ##########################################################
    # Function: _stage_1_learning
    # Function: _stage_2_learning
    # Function: _stage_3_learning

    def fit(self, dataset: ndarray) -> object:
        """
        Fitting data via the *MLC-LiNGAM* causal discovery algorithm:

        - **Stage-I**: Utilize the constraint-based method to learn a **causal skeleton**.
        - **Stage-II**: Identify the causal directions by conducting **regression** and **independence tests**
        on the adjacent pairs in the causal skeleton.
        - **Stage-III**: Detect the latent confounders with the help of the **maximal clique patterns**
        raised by the latent confounders,
        and uncover the causal structure with latent variables.

        Parameters:
            dataset:
                The observational dataset shown as a matrix or table,
                with a format of "sample (n) * dimension (d)."
                (input as Pandas dataframe is also acceptable)

        Returns:
            self:
                Update the ``adjacency_matrix`` represented as an estimated causal graph.
                The ``adjacency_matrix`` is a (d * d) numpy array with 0/1 elements
                characterizing the causal direction.
        """

        # stage-1: causal skeleton reconstruction(PC-stable algorithm)
        self._stage_1_learning(dataset)

        graph_pattern_manager = GraphPatternManager(init_graph=self._skeleton)

        # stage-2: partial causal orders identification
        self._stage_2_learning(graph_pattern_manager)

        graph_pattern_manager.store_last_managing_adjacency_matrix()

        # stage-3: latent confounders' detection
        self._stage_3_learning(graph_pattern_manager)

        return self

    # ### AUXILIARY COMPONENT(S)
    ↪   ###########################################################
    # Class: HybridFrameworkBase

    def _stage_1_learning(self, dataset: ndarray) -> object:
        """
        **Stage-I**: Causal skeleton construction (based on the PC-stable algorithm).

        Stage-I begins with a complete undirected graph and performs **conditional
        independence tests** to delete the edges between independent variables pairs,
        reducing the computational cost of subsequent regressions and independence
        ↪   tests.

        Parameters:
            dataset:
                The observational dataset shown as a matrix or table,
                with a format of "sample (n) * dimension (d)."
                (input as Pandas dataframe is also acceptable)

        Returns:
            self:
                Update `_skeleton` as the estimated undirected graph corresponding to
                the causal graph, initialize `_adjacency_matrix` via a copy of
                ↪   `_skeleton`,
                and record `_stage1_time` as the stage-1 computational time.
        """

        self._causal_skeleton_learning(dataset)

        return self

    # ### AUXILIARY COMPONENT(S)
    ↪   ###########################################################
    # Class:    GraphPatternManager
    # Function: _algorithm_2

    def _stage_2_learning(self, graph_pattern_manager) -> object:
        """
        **Stage-II**: Partial causal order identification.

        Based on the causal skeleton by stage-I,
        stage II in *MLC-LiNGAM* identifies causal directions among the adjacent
        ↪   variables
        that are implied by the skeleton.
        Causal orders relative to all variables can be partially determined by
        ↪   **regression
        and independence tests**, if variables that are relatively exogenous or endogenous
        do not be affected by latent confounders.

        Parameters:
            graph_pattern_manager:
                An auxiliary module embedded in the MLC-LiNGAM algorithm,
                managing adjacency matrices amidst the procedure between causal skeleton
                learning and causal direction orientation.

        Returns:
            self:
                Update `_adjacency_matrix` as the estimated (partial) directed acyclic
                graph (DAG) corresponding to the causal graph,
                and `record _stage2_time` as the `stage-2` computational time.
        """

        start = time.perf_counter()

        # Reconstruction of the causal skeleton entails specific pairs of adjacent variables,
        # rather than all pairs of variables.
        causal_skeleton = self._skeleton

        # MLC-LiNGAM performs regression and independence tests efficiently
        # based on the adjacency set.
        adjacent_set = GraphPatternManager.find_adjacent_set(
            causal_skeleton=causal_skeleton
        )

        # Apply Algorithm-2 (given by the MLC-LiNGAM algorithm).
        self._algorithm_2(
            corresponding_adjacent_set=adjacent_set,
            corresponding_dataset=cp.copy(self._dataset),
            corresponding_variables=np.arange(self._dim),
            graph_pattern_manager=graph_pattern_manager
        )

        # Record computational time.
        end = time.perf_counter()
        self._stage2_time = end - start

        return self

    # ### AUXILIARY COMPONENT(S) ##########################################################
    # Class:    GraphPatternManager
    # Function: _algorithm_2

    def _stage_3_learning(self, graph_pattern_manager) -> object:
        """
        **Stage-III**: Latent confounders' detection

        Stage-III will learn more causal orders if some variables are not affected
        by the latent confounders but are in the remaining subset.
        Meanwhile, the stage-III learning makes use of the causal skeleton information
        to reduce the testing space of remaining variables from all subsets to typical
        **maximal cliques**.

        Notice that the maximal cliques, including the undirected relations that cannot
        be determined, are possibly formed by latent confounders. This in turn provides
        insight to detect the latent confounders, and uncover the causal relations
        among observed and latent variables.

        Parameters:
            graph_pattern_manager:
                An auxiliary module embedded in the MLC-LiNGAM algorithm,
                featuring the algorithmic behavior of the maximal-cliques pattern
                ↪  recognition.

        Returns:
            self:
                Update ``_adjacency_matrix`` as the estimated (partial) directed
                ↪  acyclic
                graph (DAG) corresponding to the causal graph,
                ``_latent_confounder_detection`` as the undirected maximal cliques
                ↪  after
                 stage-III learning, and `record _stage3_time` as the `stage-3`
                 computational time.
        """

        start = time.perf_counter()

        # Recognize the maximal-clique pattern based on the causal skeleton.
        maximal_cliques_completely_undetermined = (
            GraphPatternManager.recognize_maximal_cliques_pattern(
                causal_skeleton=self._skeleton,
                adjacency_matrix=self._adjacency_matrix
            )
        )

        # Setup of regression, referring to MLC-LiNGAM default settings.
        regressor = LinearRegression()
        residuals_dataset = cp.copy(self._dataset)

        # Replace the variables in the clique with their corresponding residuals via
        # regressing out the effect of their confounded parents that are outside the
        ↪  clique.
        for maximal_clique in maximal_cliques_completely_undetermined:
            # Record: Each of the variable requires a single replacement if necessary.
            variables_replaced = {}
            for variable in maximal_clique:
                variables_replaced[variable] = set()

            # Get undetermined pairs within a clique.
            for i in maximal_clique:
                for j in maximal_clique[maximal_clique.index(i) + 1:]:
                    parents_i = graph_pattern_manager.managing_parents_set[i]
                    parents_j = graph_pattern_manager.managing_parents_set[j]

                    # Conduct residuals replacement if the variables share the same parents.
                    if (parents_i & parents_j) != set():
                        confounded_parents = parents_i & parents_j

                        for confounder in confounded_parents:
                            data_confounder = residuals_dataset[:, confounder]

                            if confounder not in variables_replaced[i]:
                                variables_replaced[i].add(confounder)

                                data_i = residuals_dataset[:, i]
                                residuals_i = get_residuals_scm(
                                    explanatory_data=data_confounder,
                                    explained_data=data_i,
                                    regressor=regressor
                                )
                                residuals_dataset[:, i] = residuals_i.squeeze()

                            if confounder not in variables_replaced[j]:
                                variables_replaced[j].add(confounder)

                                data_j = residuals_dataset[:, j]
                                residuals_j = get_residuals_scm(
                                    explanatory_data=data_confounder,
                                    explained_data=data_j,
                                    regressor=regressor
                                )
                                residuals_dataset[:, j] = residuals_j.squeeze()

        # Apply Algorithm-2 on the maximal cliques.
        for maximal_clique in maximal_cliques_completely_undetermined:
            # Get adjacent set with respect to the variables within maximal cliques.
            adjacent_set_clique = {}
            for variable in maximal_clique:
                adjacent_set_clique[variable] = set(maximal_clique) - {variable}

            # Apply Algorithm-2 (given by the MLC-LiNGAM algorithm).
            self._algorithm_2(
                corresponding_adjacent_set=adjacent_set_clique,
                corresponding_dataset=residuals_dataset,
                corresponding_variables=np.array(maximal_clique),
                graph_pattern_manager=graph_pattern_manager,
                _specify_adjacency=True,
                _adjacent_set=adjacent_set_clique
            )

        # Update latent confounder detection
        graph_pattern_manager.store_last_managing_adjacency_matrix()
        self._latent_confounder_detection = (
            graph_pattern_manager.get_undetermined_cliques(
                maximal_cliques=maximal_cliques_completely_undetermined
            )
        )

        # Record computational time.
        end = time.perf_counter()
        self._stage3_time = end - start

        return self

    # ### SUBORDINATE COMPONENT(S) #######################################################
    # Function: MLCLiNGAM -> _stage_2_learning
    # Function: MLCLiNGAM -> _stage_3_learning

    def _algorithm_2(
            self,
            corresponding_adjacent_set: dict,
            corresponding_dataset: ndarray,
            corresponding_variables: ndarray,
            graph_pattern_manager,
            _specify_adjacency=False,
            _adjacent_set=None
    ) -> object:
        """
        Implementation of the module "Algorithm-2" in the MLC-LiNGAM algorithm.
        """

        # ================================ INITIALIZATION =================================

        # Initialize the dataset and the relative variable set.
        adjacent_set = copy_and_rename(corresponding_adjacent_set)
        _X = copy_and_rename(corresponding_dataset)
        _x = copy_and_rename(corresponding_variables)

        # Order list for the sequential search of exogenous variables and leaf variables
        k_head = []
        k_tail = []

        # Setup of regression and independence tests, referring to MLC-LiNGAM default settings.
        regressor = LinearRegression()
        ind_test_method = 'kernel_ci'

        # ========================= IDENTIFY EXOGENOUS VARIABLES
        ↪  ==========================

        # Development notes: In accord with the pseudocode in MLC-LiNGAM:
        #   x_i or i: refer to the exogenous variable

        # Perform up-down search targeting at exogenous variables.
        repeat = True
        while repeat:

            # The last remaining variable is endogenous respectively.
            if len(k_head) == (len(_x) - 1):
                break

            # Development notes: An addition loop is combined to search the most
            # exogenous variable (to strengthen the MLC-LiNGAM algorithm).

            # Search for the most exogenous variable based on relative p-values.
            p_values_x_all = {}
            for x_i in (set(_x) - set(k_head)):

                # Get adjacent set of the candidate variable.
                adjacent_set_i = adjacent_set[x_i]

                # Check if the variable x_i is in form of a trivial sub-graph.
                if len(adjacent_set_i) == 0:
                    k_head.append(x_i)
                    continue

                # Exclude the ones in K-head-list in which
                # regressing and supplanting other variables with residuals have been
                ↪  performed.
                adjacent_set_i = adjacent_set_i - set(k_head)

                # Check if the variables are respectively the most exogenous.
                if len(adjacent_set_i) == 0:
                    k_head.append(x_i)
                    continue

                # Separately regress on adjacent variables of the candidate variable
                ↪  x_i
                # and check if all residuals are independent of it.
                p_values_x_i = []
                for x_j in adjacent_set_i:
                    residuals = get_residuals_scm(
                        explanatory_data=_X[:, x_i],
                        explained_data=_X[:, x_j],
                        regressor=regressor
                    )

                    p_value = conduct_ind_test(
                        explanatory_data=_X[:, x_i],
                        residuals=residuals,
                        ind_test_method=ind_test_method
                    )

                    p_values_x_i.append(p_value)

                # Check if the candidate variable satisfying exogeneity.
                if np.min(p_values_x_i) >= self.pc_alpha:
                    p_values_x_all[x_i] = np.min(p_values_x_i)

            # End if none of the candidate variable satisfying exogeneity.
            if len(p_values_x_all.values()) == 0:
                repeat = False

            else:
                # Mark continuous searching.
                repeat = True

                # Determine the most exogenous variable.
                p_value_max = cp.copy(self.pc_alpha)
                x_exogenous = None
                for x_i, p_value in p_values_x_all.items():
                    if p_value > p_value_max:
                        p_value_max = p_value
                        x_exogenous = x_i

                # Append the exogenous variable sequentially to k-head-list.
                k_head.append(x_exogenous)

                # Regress and supplant other variables with the residuals
                # regressed by the exogenous variable.
                for x_j in (adjacent_set[x_exogenous] - set(k_head)):
                    supplanting_residuals = get_residuals_scm(
                        explanatory_data=_X[:, x_exogenous],
                        explained_data=_X[:, x_j],
                        regressor=regressor
                    )

                    # Development notes: Residuals for supplanting are additionally
                    ↪   computed
                    # to save memory.
                    _X[:, x_j] = supplanting_residuals.ravel()

        # ============================ IDENTIFY LEAF VARIABLES
        ↪   ============================

        # Development notes: In accord with the pseudocode in MLC-LiNGAM:
```

```python
        #   x_j or j: refer to leaf variable

        # Perform bottom-up search targeting at leaf variables
        # if the causal order presents more than two variables staying undetermined.
        if len(k_head) < (len(_x) - 2):

            repeat = True
            while repeat:

                # The last remaining variable is endogenous respectively.
                if len(k_head) + len(k_tail) == (len(_x) - 1):
                    break

                # Development notes: An addition loop is combined to search the most
                # endogenous (leaf) variable (to strengthen the MLC-LiNGAM algorithm).

                # Search for the most endogenous variable based on relative p-values.
                p_values_x_all = {}
                for x_j in (set(_x) - (set(k_head) | set(k_tail))):

                    # Get adjacent set of the candidate variable.
                    adjacent_set_j = adjacent_set[x_j]

                    # Exclude ones in K-head-list in which
                    # regressing and supplanting residuals have been performed.
                    adjacent_set_j = adjacent_set_j - set(k_head)

                    # Ignore the ones in K-tail-list that are explained variables relative to x_j.
                    adjacent_set_j = adjacent_set_j - set(k_tail)

                    # Check if the variables are respectively the most exogenous.
                    if len(adjacent_set_j) == 0:
                        # k_tail.insert(0, x_j)
                        k_head.append(x_j)
                        continue

                    #  Regress the candidate variable x_j on all its adjacent variables
                    #  and check if its residuals are all independent of them.
                    residuals = get_residuals_scm(
                        explanatory_data=_X[:, list(adjacent_set_j)],
                        explained_data=_X[:, x_j],
                        regressor=regressor
                    )
                    p_value = conduct_ind_test(
                        explanatory_data=_X[:, list(adjacent_set_j)],
                        residuals=residuals,
                        ind_test_method=ind_test_method
                    )
                    p_values_x_j = copy_and_rename(p_value)

                    # Check if the candidate variable is likely the leaf variable.
                    if p_values_x_j >= self.pc_alpha:
                        p_values_x_all[x_j] = p_values_x_j

                # End if none of the candidate variable is likely the leaf variable.
                if len(p_values_x_all.values()) == 0:
                    repeat = False

                else:
                    # Mark continuous searching.
                    repeat = True

                    # Determine the most endogenous variable.
                    p_value_max = cp.copy(self.pc_alpha)
                    x_leaf = None
                    for x_j, p_value in p_values_x_all.items():
                        if p_value > p_value_max:
                            p_value_max = p_value
                            x_leaf = x_j

                    # Insert the leaf variable at the top of k-tail-list.
                    k_tail.insert(0, x_leaf)

        # ======================== IDENTIFY PARTIAL CAUSAL ORDER ==========================

        # Update causal skeleton to partial causal structure according to
        # K-Head and K-Tail list.
        graph_pattern_manager.identify_partial_causal_order(
            k_head=k_head,
            k_tail=k_tail
        )

        self._adjacency_matrix = graph_pattern_manager.managing_adjacency_matrix
        self._parents_set = graph_pattern_manager.managing_parents_set

        return self

    @property
    def latent_confounder_detection_(self):
        return self._latent_confounder_detection
```