Title

Nonlinear Causal Discovery via Sequential Edge Orientation for Directed Acyclic Graphs and Mixed Graphs

ISBN

9798247928423

Author

Huang, Stella

Publication Date

2026-05-29

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Nonlinear Causal Discovery via Sequential Edge Orientation for Directed Acyclic Graphs and Mixed Graphs

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Statistics

by

Stella Huang

2026

ABSTRACT OF THE DISSERTATION

Nonlinear Causal Discovery via Sequential Edge Orientation
for Directed Acyclic Graphs and Mixed Graphs

by

Stella Huang
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2026
Professor Qing Zhou, Chair

Recent advances in causal learning have established the identifiability of a directed acyclic graph (DAG) under additive noise models (ANMs), spurring the development of various causal discovery methods. However, most existing methods make restrictive model assumptions, rely heavily on general independence tests, or require substantial computation.

To address these limitations, we propose a sequential procedure to orient undirected edges in a completed partial DAG (CPDAG), representing an equivalence class of DAGs, by leveraging a pairwise additive noise model (PANM) to identify their causal directions. We prove that this procedure can recover the true causal DAG assuming a restricted ANM. Building on this result, we develop a novel constraint-based algorithm for learning causal DAGs under nonlinear ANMs. Given an estimated CPDAG, we develop a ranking procedure that sorts undirected edges by their adherence

to the PANM, which defines an evaluation order of the edges. To determine the edge direction, we devise a statistical test that compares the log-likelihood values, evaluated with respect to the competing directions, of a sub-graph comprising just the candidate nodes and their identified parents in the partial DAG. We further establish the structural learning consistency of our algorithm in the large-sample limit. Extensive experiments on synthetic and real-world data sets demonstrate that our method is computationally efficient, robust to model misspecification, and consistently outperforms many existing nonlinear DAG learning methods.

We also study nonlinear causal structure learning in the presence of latent variables, where the latent projection of a DAG is represented by an acyclic directed mixed graph (ADMGs). Existing approaches are often limited by model assumptions and rely on iterative testing. To this end, we develop a sequential edge orientation framework for determining undetermined edges in a partial ancestral graph (PAG) that combines the strengths of constraint-based and score-based methods. We propose a graphical criterion for identifying edges whose orientations can be correctly inferred through conditional independence tests, enabling targeted edge evaluation rather than repeated global testing. For the remaining edges, we directly compare competing orientations using log-likelihood estimates obtained through a parameter estimation procedure tailored to nonlinear models that more accurately recovers the covariance structure. Through extensive simulations, we demonstrate that the proposed framework substantially improves the recovery of both causal and latent confounding relationships and achieves greater accuracy than existing competing methods.

The dissertation of Stella Huang is approved.

Yingnian Wu

Oscar Madrid-Padilla

Arash A. Amini

Qing Zhou, Committee Chair

University of California, Los Angeles

2026

Dedicated to my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Over the years, I have become acutely aware of the scarcity of time and space, often prioritizing efficiency and outcomes over the experience of the journey itself. My Ph.D. studies, however, has afforded me the uncommon gift of time: the freedom to think deeply without interruptions and pursue meaningful problems with care and curiosity. In doing so, it has taught me the value of sustained effort and intellectual exploration. I am deeply grateful for the opportunity to contribute to the field of statistics and for the remarkable people who accompanied me along the way.

First and foremost, I would like to thank my advisor, Professor Qing Zhou, for his mentorship and guidance throughout my doctoral studies. He has dedicated countless hours to discussing research ideas, provided direction during difficult moments, and carefully reviewed our work with remarkable attention to detail. I deeply admire his commitment to pursuing meaningful and rigorous research, viewing publications not as the goal, but as a natural outcome of impactful work. I often left our meetings feeling inspired and intellectually energized; I have learned immensely from both his mentorship and his example as a scholar. I would also like to thank my committee members, Professors Arash Amini, Oscar Madrid-Padilla, and Yingnian Wu, for their valuable feedback on my work.

I am grateful to have shared my interest in causal learning with such supportive and thoughtful labmates. I thank Jireh Huang, Gabriel Ruiz, and Stephen Smith for generously sharing their knowledge, and always answering my questions when I first joined the lab in 2021. I am especially thankful to Dale Kim for our many long and insightful conversations on latent graphical models, which enriched both my understanding of the field and my perspective on research more broadly. I also thank

Alex Chen and Yijia Zhao, with whom I entered the program with. I particularly appreciate Yijia's positivity and encouragement, and I am grateful we were able to share similar research experiences throughout the program.

I am equally thankful for the friendships I formed during graduate school. Tanvi Shinkre and Elizabeth O'Neill were among my earliest companions in the program when we began as master's students during the pandemic. Our close communication during the remote-learning period provided reassurance and support during that uncertain time, and our friendship has continued ever since. I thank Jiayi Li and Edouardo Hoang for their encouragement and advice when I was navigating through my first end-to-end research project. I am also grateful to Andrew Lizzaraga for his kindness, empathy, intellectual curiosity, and dedication to improving both student learning and departmental culture. Sophie Phillips and Danny Ying made my days in the office so enjoyable, and I appreciate sharing the experience of marathon training alongside doctoral studies with them. I also thank Jasen Zhang and Kyle Wu for initiating plans offline and for their empathy. Christina Rodriguez always brightened my day with stories to share and thoughtful gestures celebrating even my small accomplishments. And thank you to Joseph Resch, Davis Berlind, Minglu Zhao, Aishni Parab, and Duy Pham for their friendship and support too. I feel fortunate to have been surrounded by intellectually curious and kind-hearted peers.

Beyond graduate school, I am grateful to have maintained longstanding friendships while forming new ones throughout this journey. I especially thank my close friend Eyckie Zheng, who was only a 10 minute drive away, for her companionship and care from our freshman year of college to now. I am also deeply thankful to David Zhao for his patience, honesty, and unwavering support since the beginning of my undergraduate years; thank you for keeping me grounded. I additionally thank my

run clubs for a community rooted in a shared passion and dedication towards the sport.

Finally, I would like to thank my family, whose love and support have been the foundation of my confidence in countless decisions and challenges, including the pursuit of this degree. My grandfather instilled in me the values of discipline and academic excellence, while my grandmother has always shown me unconditional love and care. I am grateful to my brother for his reliability and steadfast support throughout the years. I also feel fortunate that we were both able to pursue graduate degrees while living in Los Angeles and are graduating in the same year. Most of all, I thank my mother. Her love, patience, wisdom, and unwavering belief in me shaped every stage of this journey. Throughout my studies, she was always willing to listen, and encouraged me to pursue what was meaningful rather than what was easy. In particular, I thank her for always telling me what I need to hear, rather than what I want to hear. I am deeply grateful for her sacrifices, her example, and her constant support.

VITA

2020–2026 Ph.D. Candidate in Statistics, University of California, Los Angeles

2016–2020 B.S. Mathematics of Computation, minor in Statistics, University of California, Los Angeles

PUBLICATIONS

Huang, Stella, and Qing Zhou. "Nonlinear Causal Discovery through a Sequential Edge Orientation Approach." arXiv preprint arXiv:2506.05590, 2025.

CHAPTER 1

Introduction

1.1 Causal Bayesian Networks

Structural causal models [Pearl, 2000] represent the causal relations amongst a set of variables using a directed acyclic graph (DAG), while the underlying data generating process is described by a set of structural equation models (SEMs). In practice, the true DAG is often unknown or difficult to construct due to limited domain knowledge. Consequently, a wide range of causal discovery methods have been developed to learn the underlying DAG or its equivalence class from observational data [Glymour et al., 2019, Vowels et al., 2022].

In a general SEM, a variable is modeled as a deterministic function of other variables and an exogenous noise term. More precisely, for random variables $\{X_i\}_{i=1}^p$ with corresponding error terms $\{\varepsilon_i\}_{i=1}^p$, the general SEM takes the form

$$X_i = f_i(\mathrm{PA}_i, \varepsilon_i), \qquad i = 1, \ldots, p, \tag{1.1}$$

where $\mathrm{PA}_i = \{X_j : j \in \mathrm{pa}_{\mathcal{G}}(i)\}$ denotes the parent set of X_i.

A graph $\mathcal{G} = (V, E)$ consists of a set of vertices $V = [p] := \{1, \ldots, p\}$ and a set of edges $E \subseteq V \times V$. For a pair of distinct nodes $i, j \in V$, a directed edge $i \to j \in E$ indicates node i is a parent to its child node j. The parent and children sets of i

in $\mathcal{G}$ are respectively denoted as $\text{pa}_{\mathcal{G}}(i)$ and $\text{ch}_{\mathcal{G}}(i)$. In contrast, there may exist an undirected edge $i - j \in E$ in $\mathcal{G}$, where i is called a neighbor of j, i.e. $i \in \text{ne}_{\mathcal{G}}(j)$, and vice versa. A *directed acyclic graph* (DAG) $\mathcal{G}$ consists only of directed edges and does not admit any directed cycles. A related type of graph is the *partially directed acyclic graph* (PDAG), which contains both directed and undirected edges but does not contain any cycles in its directed subgraphs.

1.1.1 Directed Acyclic Graphs

A *causal DAG* is a structural causal model that employs a DAG $\mathcal{G}$ to represent causal relations among random variables $X = \{X_1, \ldots, X_p\}$. Specifically, these causal relations are described by SEMs of the form in (1.1), where $\text{PA}_i = \{X_j : j \in \text{pa}_{\mathcal{G}}(i)\}$. The probability distribution over the noise variables $p(\varepsilon) = \prod_{i=1}^{p} p(\varepsilon_i)$ induces a distribution $p(X)$ over $\{X_i\}_{i=1}^{p}$. In particular, the joint distribution $p(X_1, \ldots, X_p)$ satisfies the *Markov condition* as its density factorizes according to $\mathcal{G}$:

$$p(X_1, \ldots, X_p) = \prod_{i=1}^{p} p(X_i | \text{PA}_i), \tag{1.2}$$

where $p(X_i|\text{PA}_i)$ denotes the density of X_i conditional on its parent set. The Markov condition implies that any X_i is independent of its non-descendant nodes given its parents PA_i. Hereafter, we identify the nodes V and the random variables X.

The Causal Markov condition also presents a means for inference on the conditional independence relations described by $p(X)$ according to the structure in $\mathcal{G}$. To infer the structure of the graph $\mathcal{G}$ from observed data, we require the assumptions of *causal sufficiency* and *faithfulness* to hold [Pearl, 2000]. *Causal sufficiency* is satisfied when all common causes of any distinct pair of nodes are observed, resulting in no latent confounders. Suppose $A, B, C \subset [p]$ are disjoint subsets. A distribution P and DAG

$\mathcal{G}$ are *faithful* to each other if the conditional independence relations in P have a one-to-one correspondence with the d-separation relations in $\mathcal{G}$:

$$A \perp\!\!\!\perp B|C \text{ in } P \iff C \text{ d-separates } A, B \text{ in } \mathcal{G} \tag{1.3}$$

Faithfulness also implies the causal minimality condition. The pair $(\mathcal{G}, P)$ satisfies the *causal minimality* condition if P is not Markov to any proper subgraph of $\mathcal{G}$ over the vertex set V [Spirtes et al., 2000]:

$$X_i \not\perp\!\!\!\perp X_j \mid \mathrm{PA}_i \setminus \{X_j\}, \quad \forall X_j \in \mathrm{PA}_i. \tag{1.4}$$

An equivalent statement is that there is no variable that is conditionally independent of any of its parents, given the remaining parent set.

1.1.2 Markov Equivalence Class

Two DAGs are *Markov equivalent* if and only if they have identical skeletons and v-structures, which are ordered triplets of nodes i, j, k oriented as $i \to k \leftarrow j$ with no edge between i, j [Verma and Pearl, 1990]. Markov equivalent DAGs encode the same set of d-separations and form an equivalence class. Without further constraints on the function class in (1.1), we can only identify their equivalence class. The Markov equivalence class (MEC) is represented by a *completed partially directed acyclic graph* (CPDAG) $\mathcal{E}$, a PDAG with specific structural properties [Andersson et al., 1997]. A CPDAG satisfies the following conditions:

1. G is a chain graph.
2. G_τ is chordal for every chain component τ of G.
3. The configuration $a \to b - c$ does not occur as an induced subgraph of G.

4. Every arrow $a \to b$ is strongly protected.

Chain graphs are PDAGs for which the vertex set V can be partition as $V = V_1 \cup ... \cup V_T$. As such, all edges for nodes in the same chain component V_t are undirected and all edges between two different chain components V_s, V_t are directed. In the CPDAG, a set of nodes connected by undirected edges form a chain component. The last statement requires that every directed edge is compelled, or strongly protected, and every undirected edge is reversible in the CPDAG.

The CPDAG $\mathcal{E}$ of a DAG $\mathcal{G}$ is typically obtained by first identifying v-structures in the skeleton, and then applying *Meek's rules*, a set of four rules that orient edges based on graphical patterns [Meek, 1995]. Meek's rules identify additional directed edges in the graph without introducing new v-structures. A *maximally oriented PDAG* is a PDAG for which no edges can be further oriented by Meek's rules. As an example, a CPDAG $\mathcal{E}$ is a maximally oriented PDAG, or a maximal PDAG for short.

A *consistent extension* of a PDAG $\mathcal{G}$ is a DAG, $\widetilde{\mathcal{G}}$, obtained by orienting all undirected edges in $\mathcal{G}$ without introducing new v-structures. Therefore, $\widetilde{\mathcal{G}}$ has the same skeleton, the same orientations of all directed edges in $\mathcal{G}$, and the same v-structures as $\mathcal{G}$ [Dor and Tarsi, 1992]. While not all PDAGs can be extended to a DAG, a CPDAG $\mathcal{E}$ is extendable since each DAG in the equivalence class represented by $\mathcal{E}$ is a consistent extension of $\mathcal{E}$. A DAG may be obtained from a CPDAG by iteratively making edge orientations without introducing new v-structures and applying Meek's rules, while preserving the compelled edges [Wienöbst et al., 2021].

1.2 Structural Learning

1.2.1 Causal Learning Methods

The estimation of a DAG from data is a non-trivial and computationally intensive task due to its enormous search space. In particular, the number of candidate DAGs is super exponential in the number of nodes Robinson [2006]. Under certain search strategies and moderate number of variables, there are quite some successful causal learning methods. In general, causal learning methods can be classified as one of the three following classes [Zhou, 2025]. Constraint based methods first utilize conditional independence tests to determine the existence of an edge between a pair of nodes, thereby obtaining a skeleton and then inferring the edge orientation. Score-based methods aim to optimize a scoring criterion on graphs that reflects the goodness-of-fit of the structure to the data, often using the BIC score or Bayesian metrics, with the GES algorithm being a representative example Chickering [2002b]. Last, optimization-based methods have reformulated the structural learning task as a continuous-optimization problem by devising an algebraic characterization of DAGs; methods of this class, like NOTEARS Zheng et al. [2018] and DAGMA Bello et al. [2022], use deep-learning methods to perform optimization and model the SEMs.

The PC algorithm [Spirtes et al., 2000] is the most prominent constraint-based algorithm and regarded as a gold standard. Given a complete graph over p nodes, the algorithm first tests the conditional independence statement $X_i \perp\!\!\!\perp X_j \mid X_S, S \subseteq (V \setminus \{i, j\})$ for increasing size of S. Under the assumption of faithfulness, the edge $i - j$ is deleted from the graph if X_i and X_j are conditionally independent. Common metrics for CI test include partial correlation, mutual information, and, recently

more popular, kernel-based test statistics Zhang et al. [2012] for detecting nonlinear dependencies. With the skeleton learned, the algorithm searches for v-structures amongst a triplet (i, j, k) and orients the configuration as $i \to k \leftarrow j$ if $X_i \perp\!\!\!\perp X_j \mid X_k$. Last, it applies Meek's rules to further orient any undirected edges to obtain the CPDAG of a DAG. Given the conditional independence oracle for all pairs of nodes, the CPDAG can be correctly recovered [Colombo and Maathuis, 2014].

1.2.2 Identifiability of the DAG

A common assumption regarding the SEM (1.1) is that each $f_i(\cdot)$ is a linear function with additive Gaussian noise. Although analytically convenient, this assumption is not only unrealistic, but also limits algorithms to learn the CPDAG, rather than the exact DAG Chickering [2002b]. As demonstrated in Hoyer et al. [2008], the joint density under a simple linear, Gaussian model $y = f(x) + \varepsilon_y$, for some linear function f with independent noise term ε_y, is symmetric with respect to both the predictor and outcome variables. This implies that a backward model $x = g(y) + \varepsilon_x$, where g is another linear function, would fit equally well to the data. Nevertheless, previous research has demonstrated conditions enabling the identifiability of the true DAG from observational data. Initial works by Hoyer et al. [2008] and Shimizu et al. [2006] proved bivariate identifiability under nonlinear functions and/or non-Gaussian noises, respectively. These assumptions break the symmetry in the bivariate distribution of two nodes, enabling the identification of causal directions [Zhang and Hyvärinen, 2016].

For the nonlinear case, Hoyer et al. [2008] show that the causal direction can be determined under the additive noise model (ANM). Under the ANM, each variable

X_i is a function of its parent nodes PA_i in DAG $\mathcal{G}_0$ plus an independent additive noise ε_i, i.e.

$$X_i = f_i(\text{PA}_i) + \varepsilon_i, \tag{1.5}$$

where f_i is an arbitrary function. It is assumed that the noise variables $\{\varepsilon_i\}_{i=1}^p$ are jointly independent and independent of parent nodes, i.e. $\varepsilon_i \perp\!\!\!\perp \text{PA}_i$. Throughout this work, we assume causal minimality for the ANM, which is satisfied as long as the function f_i, for all i, is not constant in any of its arguments [Peters et al., 2014]. Let ND_j denote the non-descendants of X_j in the underlying DAG and $L(Y)$ denote the distribution of a random variable Y. Specifically, Hoyer et al. [2008] have proven that if the triple $(f_j, L(X_i), L(N_j))$ satisfies the following condition, then the causal relation is identifiable.

Condition 1. Let p_{X_i} and p_{N_j} be strictly positive densities of $L(X_i)$ and $L(N_j)$, respectively. The triple $(f_j, L(X_i), L(N_j))$ does not solve the following differential equation for all x_i, x_j with $\nu''(x_j - f(x_i))f'(x_i) \neq 0$:

$$\xi''' = \xi''\left(-\frac{\nu'''}{\nu''}f' + \frac{f''}{f'}\right) - 2\nu''f''f' + \nu'f''' + \frac{\nu'\nu'''f''f'}{\nu''} - \frac{\nu'(f'')^2}{f'}, \tag{1.6}$$

where $f := f_j$, $\xi := \log p_{X_i}$, and $\nu := \log p_{N_j}$. To improve readability, we have omitted the arguments $x_j - f(x_i), x_i$, and x_i for ν, ξ, and f and their derivatives, respectively.

We turn our attention to the identifiabiliy of the DAG. A restricted additive noise model is an ANM with restrictions on the functions f_i, conditional distributions of X_i, and noise variables. Peters et al. [2014] then utilize this result to prove the identifiability of a DAG assuming a restricted additive noise model.

Definition 1. We call the SEM (1.5) a restricted additive noise model if for all $j \in V = [p]$, $i \in \mathrm{PA}_j$ and all sets $S \subseteq V$ with $\mathrm{PA}_j \setminus \{i\} \subseteq S \subseteq ND_j \setminus \{i, j\}$, there is an x_S with $p_S(x_S) > 0$, s.t.

$$\left(f_j(x_{\mathrm{PA}_j \setminus \{i\}}, \underbrace{\cdot}_{X_i}), L(X_i \mid X_S = x_S), L(N_j) \right)$$

satisfies Condition 1. In particular, we require the noise variables to have non-vanishing densities and the functions f_j to be continuous and three times continuously differentiable.

Under the causal minimality assumption, a key result is that the true DAG $\mathcal{G}_0$ can be identified from the joint distribution $p(X_1, \ldots, X_p)$ when the SEM for $\{X_i\}_{i=1}^p$ satisfies a restricted additive noise model.

We also consider the *causal additive model* (CAM), a special case of the ANM where the function f_i is additive [Bühlmann et al., 2014]. The model is defined as

$$X_i = \sum_{j \in \mathrm{pa}(i)} f_{i,j}(X_j) + \varepsilon_i, \qquad i = 1, \ldots, p, \tag{1.7}$$

where $f_{i,j}$ are three times differentiable, nonlinear functions and the error terms $\varepsilon_i \sim N(0, \sigma_i^2)$ independently with $\sigma_i^2 > 0$. Peters et al. [2014] have demonstrated that the true DAG $\mathcal{G}_0$ can be identified from the joint distribution $p(X_1, \ldots, X_p)$ when the SEM for $\{X_i\}_{i=1}^p$ satisfies a CAM. Furthermore, the source nodes are also allowed to have a non-Gaussian density. As the true SEM under a general ANM is difficult to recover in a practical setting, assuming a CAM assists in better recovering the underlying causal relations, by utilizing regression models such as the generalized additive model.

Let $\mathcal{F}$ be a class of three times differentiable univariate functions that is closed with respect to the $\mathcal{L}_2$ norm of $P(X_i), i = 1, \ldots, p$. We define a space of additive functions

$$\mathcal{F}^{\oplus k} := \left\{ f : \mathbb{R}^k \to \mathbb{R}, f(x) = \sum_{i=1}^{k} f_i(x_i), \quad f_i \in \mathcal{F} \right\}, \quad k \in [p]. \tag{1.8}$$

1.3 Overview

The remainder of this dissertation is organized as follows. In Section 2, we develop a novel causal discovery algorithm for nonlinear DAG learning by sequentially orienting undirected edges in a CPPAG to recover the DAG. We establish theoretically that, at any stage of the learning process, there exists at least one undirected edge that can be correctly oriented, then we introduce the pairwise additive noise model criterion (PANM) to characterize such edges. Building on this result, we propose a sequential edge orientation framework that ranks undirected edges using a metric derived from the PANM criterion and determines causal direction through a likelihood-based statistical test.

Chapter 3 presents the empirical evaluation of this framework on both synthetic and real-world datasets. Across settings where model assumptions are satisfied as well as violated, our method demonstrates strong causal learning accuracy and computational efficiency relative to competing approaches. We further examine the contribution of each stage and component of the orientation procedure individually, alongside their combined performance across diverse scenarios, to assess robustness and practical effectiveness.

In Chapter 4, we extend nonlinear causal discovery to settings with latent variables

present. We develop a graphical criterion for distinguishing between two classes of undetermined edges and propose specialized orientation procedures for each using constraint-based and score-based strategies separately. In particular, the score-based framework enables direct comparison between directed and bidirected edges using the bivariate conditional log-likelihood estimates, opposed to the log-likelihood of the full graph. We derive the corresponding score function and present a general framework for computing these log-likelihood estimates. Finally, Chapter 5 summarizes the dissertation's contributions and discusses promising directions for future research.

CHAPTER 2

Nonlinear Causal Discovery through a Sequential Edge Orientation Approach

2.1 Motivation and Overview

For the general SEM (1.1), a common assumption is that each $f_i(\cdot)$ is a linear SEM with additive Gaussian noise. Although analytically convenient, this assumption is not only overly simplistic, but also limits algorithms to learning a completed partially direct acyclic graph (CPDAG), which encodes a set of Markov equivalent DAGs sharing the same conditional independence relations, rather than the exact true DAG [Chickering, 2002a]. Nevertheless, previous research has demonstrated conditions enabling the identifiability of the true DAG from observational data. Initial works by Hoyer et al. [2008] and Shimizu et al. [2006] proved bivariate identifiability under nonlinear functions and/or non-Gaussian noises, respectively. These assumptions break the symmetry in the bivariate distribution of two nodes, enabling the identification of causal directions [Zhang and Hyvärinen, 2016].

2.1.1 Relevant Work

In this paper, we focus on learning causal DAGs from nonlinear data. Prior works, such as the additive noise model (ANM) by Hoyer et al. [2008] and post-nonlinear model by Zhang and Hyvärinen [2009], have established the identifiability of the true DAG under particular assumptions on the function class $f_i(\cdot)$ in the general SEM. Peters et al. [2011] proved identifiability for the general SEM (1.1) by defining the concept of an identifiable functional model class. Specifically, the authors present a criterion on $\{X_i, f_i(\mathrm{PA}_i), \varepsilon_i\}$ for DAG identifiability, hence generalizing previous works focusing on specific models only. They also developed an algorithm to find all DAGs that satisfy their identifiability condition through iteratively testing for independence between residuals and parents.

New causal discovery algorithms have then ensued from these identifiability results. In the domain of constraint-based methods, kernel-based tests have garnered considerable attention. The kernel-based conditional independence (KCI) test, a notable early example proposed by Zhang et al. [2012], captures nonlinear relationships by mapping random variables to reproducing kernel Hilbert spaces using kernel methods. Building on this, the algorithm RESIT [Peters et al., 2014] utilizes a kernel-based statistic to recursively identify sink nodes and infer the topological ordering of a DAG. Another line of work employs regression-based methods to detect nonlinearity. For instance, the nonlinear invariant causal prediction (ICP) framework [Heinze-Deml et al., 2018] fits regression models to data collected in different environments, and then tests for differences in residual distributions across environments after interventions to detect associations. Monti et al. [2020] extend independence component analysis to the nonlinear setting and use the method to infer causal models. It utilizes non-

stationary data and nonlinear independent component analysis to identify exogenous error variables, and then performs independence tests to determine causal directions. Gretton et al. [2009] consider the cases where the nonlinear function f_i is invertible or where the noise is not additive, and demonstrate that only a PDAG is identifiable in these cases. Their method thereby focuses on identifying non-invertible ANMs in a Markov equivalence class (MEC) through iterative residual independence testing. Improving upon an estimated CPDAG, propose a statistical test that orients the undirected edges by comparing the goodness of fit of models corresponding to the two possible directions. With minimal assumptions on the regression model, this approach offers flexibility in modeling and greater applicability to real-world data.

However, these constraint-based methods also face practical limitations that hinder their widespread adoption. A significant drawback to the iterative testing approach is the computational cost, notably for kernel-based methods. As its runtime scales quadratically with sample size, the KCI test is computationally expensive to employ in constraint-based algorithms, and especially inefficient for learning larger DAGs [Strobl et al., 2019]. Additionally, the performance of the KCI test heavily depends on the chosen kernel function, which is often problem-specific and difficult to tune. The nonlinear ICP requires a sufficient number of observed environments to detect causal relations [Rosenfeld et al., 2021] and thus cannot be applied to data generated under a classical i.i.d. setting. Moreover, the accuracy of regression-based approaches is contingent on the quality of the estimated model, which requires strong domain knowledge for model selection and large sample sizes to accurately approximate the true SEM [Shah and Peters, 2020]. Model misspecification can lead to violations of key assumptions, ultimately inflating false positive rates in independence tests [Li and Fan, 2020]. More broadly, recent efforts tend to focus on developing independence tests –

without fully developing them into scalable, full-fledged causal discovery algorithms [Hasan et al., 2023] – or approximating key statistical quantities to improve test efficiency, with less emphasis on using these metrics to infer causal relations.

Score-based methods for nonlinear learning have also seen significant development in recent years. The algorithm CAM [Bühlmann et al., 2014], for example, assumes an additive noise model with Gaussian errors and maximizes the joint log-likelihood function to learn a DAG. Huang et al. [2018] propose a score function by measuring independence in a reproducing kernel Hilbert space, thereby enabling a novel approach to learn the CPDAG under nonlinear relations and arbitrary distributions. Ramsey et al. [2025] obtain a set of embeddings for each variable using a truncated set of Legendre polynomials, allowing for fast approximation of nonlinear, additive SEMs. They propose a score function based on this method to estimate nonlinear CPDAGs. Several recent works have reformulated the structural learning task as a continuous-optimization problem by devising an algebraic characterization of DAGs [Zheng et al., 2018]. Prominent examples include NOTEARS [Zheng et al., 2020], DAG-GNN [Yu et al., 2019], and DAGMA [Bello et al., 2022], which incorporate deep learning techniques to enhance the flexibility of SEM estimation. Another line of work, exemplified by the SCORE algorithm [Rolland et al., 2022], iteratively identifies leaf nodes in a DAG using the Jacobian of the score function. However, score-based methods generally heavily depend on model assumptions, require intensive computational time, and are not guaranteed to find a globally optimal structure. Some of the above deep learning-based methods, in particular, perform poorly when the data is standardized, as standardization erases causal order information from the marginal variance when minimizing the least squares objective function [Reisach et al., 2021].

2.1.2 Contribution of This Work

In this work, we propose a novel causal discovery algorithm SNOE (Sequential Nonlinear Orientation of Edges) for nonlinear DAG learning. SNOE builds upon the CPDAG learned by classical methods and sequentially determines the causal direction of undirected edges to learn the true DAG.

In lieu of inferring the causal order of nodes, we introduce a local identifiability criterion based on a pairwise additive noise model (PANM). This criterion determines whether a given undirected edge in a partially directed acyclic graph (PDAG) can be correctly oriented. Edges that fulfill this criterion are ensured to be correctly oriented in the large-sample limit, without inducing errors in subsequent orientations. We prove that, at the population level, the algorithm consistently recovers the true DAG from its CPDAG. Leveraging this result, we devise a sequential algorithm to identify undirected edges following the PANM criterion and infer their causal directions. As such, the algorithm effectively learns the DAG at a local scale by evaluating edges individually without necessitating evaluation of all nodes or the entire graph. To determine the orientation of each edge, SNOE employs a likelihood-ratio test that compares the bivariate conditional probability distributions over the sub-DAG on both nodes and their learned parent sets under the competing directions. Contrary to general independence tests, the likelihood ratio test returns a definitive decision regarding the causal direction, thereby bridging the gap between the task of detecting nonlinear conditional independence relations and the structural learning problem. Empirical results further show that the test is robust to violations of model misspecification and yields accurate results across different functional settings.

The main contributions of this work are summarized below:

- A novel criterion derived from a pairwise additive noise model to determine the identifiability and correct orientation of undirected edges in a PDAG;
- An algorithm that is guaranteed at the population level to identify the true DAG from its Markov equivalence class by orienting undirected edges, according to their adherence to the PANM criterion, in a sequential manner;
- Theoretical results for the structural learning consistency of our algorithm in the large-sample limit;
- Higher accuracy and faster computation time compared to competing nonlinear DAG learning methods.

At a conceptual level, the sequential orientation algorithm is the most significant contribution of this work. In essence, our method is rooted in this central idea: Starting from the CPDAG, there exists at least one undirected edge whose orientation can be determined by the PANM criterion at *any* iteration in our sequential algorithm, leading to the correct recovery of the true DAG. In this process, we check whether the PANM is satisfied for an undirected edge only conditional on the identified parents of the two nodes. This is made possible with a careful design of edge orientations by the PANM criterion and by a subset of the Meek's rules [Meek, 1995]. Although Gretton et al. [2009] also orients edges in a CPDAG, they use a greedy search to iteratively identify nodes satisfying a non-invertible ANM by examining all candidate parent sets of a node, i.e., by checking all subsets of its neighbors (connected to the node by an undirected edge). This is clearly different from our sequential orientation procedure that only considers identified parents without the need to examine any subset of the neighbors. The method proposed by Wang and Zhou [2021] ranks the undirected

edges in a CPDAG by a goodness-of-fit metric, without establishing its rigorous property in terms of edge orientation or recovery of the true DAG. Moreover, they assume a specific piecewise linear SEM, while we consider a more general nonlinear ANM. Peters et al. [2014] employs independence tests to identify a sink node in a sequential manner, which is distinct from the above sequential edge orientation methods.

The paper is structured as follows. In Section 2.2, we present the central notion of our work, the sequential edge orientation procedure and the finite-sample version of the algorithm. In Section 2.3, we discuss the two key components of the edge orientation step: the ranking procedure and the orientation test. Then, we establish the structural learning consistency of the algorithm in Section 2.4. Section 3.1 presents the performance of our method against competing methods on simulated data, with a detailed analysis of some intermediate results. We further demonstrate its performance in causal discovery on real-world data in Section 3.2. Last, we summarize our work and outline directions for future research in Section 2.5. Proofs and supplemental numerical results are provided in the Appendix.

2.2 Algorithm Overview

Classical causal discovery methods learn an equivalence class of the true DAG, represented by a CPDAG. Since nonlinear DAG models are identifiable [Peters et al., 2014], we aim to further infer the causal directions of undirected edges in a CPDAG or, more generally, a PDAG for the finite sample case. To this end, we formulate a novel criterion, named the *pairwise additive noise model* (PANM), to determine which undirected edges in a PDAG can be correctly oriented at the current stage

in a sequential manner. Once an edge fulfilling this criterion is oriented, we further apply additional graphical rules to orient more undirected edges. We show that this sequential procedure is able to orient all undirected edges in a CPDAG into the true causal DAG, thus accomplishing the goal of identifying all causal relations among the variables. In this section, we introduce the pairwise additive noise model, the sequential edge orientation procedure, and the finite sample version of our full algorithm.

2.2.1 Pairwise Additive Noise Model

The pairwise additive noise model, a set of SEMs defined with respect to two nodes, encapsulates the conditions sufficient for edge orientation. For an undirected edge $X - Y$ in a PDAG $\mathcal{G}$, given their parent sets $\text{pa}_{\mathcal{G}}(X)$ and $\text{pa}_{\mathcal{G}}(Y)$, its causal direction in true DAG $\mathcal{G}_0$ can be identified when (X, Y) follows a PANM. In certain cases, a pair of nodes may follow a PANM even if their parent sets in $\mathcal{G}_0$ are not fully identified, i.e. $\text{pa}_{\mathcal{G}}(v) \subset \text{pa}_{\mathcal{G}_0}(v)$. We demonstrate how this criterion lays the foundation for our edge orientation procedure.

Definition 2 (Pairwise Additive Noise Model)**.** Let X, Y be two random variables and Z_1, Z_2 be two sets of random variables. We say that $[X, Y \mid Z_1, Z_2]$ follows a *pairwise additive noise model* if either (i) or (ii) holds:

(i) $X = f_X(Z_1) + \varepsilon_X$, $\varepsilon_X \perp\!\!\!\perp Z_1$ and $Y = f_Y(X, Z_2) + \varepsilon_Y$, $\varepsilon_Y \perp\!\!\!\perp \{X, Z_2\}$,

(ii) $X = f_X(Y, Z_1) + \varepsilon_X$, $\varepsilon_X \perp\!\!\!\perp \{Y, Z_1\}$ and $Y = f_Y(Z_2) + \varepsilon_Y$, $\varepsilon_Y \perp\!\!\!\perp Z_2$.

Furthermore, we assume both SEMs above satisfy Condition 1 in Section 1.2.2 for any value (z_1, z_2) in the domain of (Z_1, Z_2).

Within the context of causal learning, the structural equation models in conditions (i) and (ii) correspond to the additive noise models under $X \to Y$ and $Y \to X$, respectively, where $Z_1 = \text{pa}_{\mathcal{G}}(X)$ and $Z_2 = \text{pa}_{\mathcal{G}}(Y)$ in PDAG $\mathcal{G}$. In this regard, we may simply say that the undirected edge $X - Y$ or the pair of nodes (X, Y) in the PDAG $\mathcal{G}$ satisfies the PANM. To determine whether (X, Y) follows a PANM, we can test if the independence relations hold. More specifically, only the PANM constructed under the correct orientation of $X - Y$ yields the independence relations between the noise term and the parent variables. The precise statement is summarized into the following lemma, which is an immediate consequence of the identifiability of bivariate ANMs [Hoyer et al., 2008].

Lemma 1. *Assume $\{X_i\}_{i=1}^p$ follows a restricted ANM with respect to a DAG $\mathcal{G}_0$. Suppose two nodes X, Y are connected by an undirected edge in a PDAG $\mathcal{G}$ which has a consistent extension to $\mathcal{G}_0$. If $[X, Y | pa_{\mathcal{G}}(X), pa_{\mathcal{G}}(Y)]$ follows the PANM, then the causal direction between X and Y is identifiable.*

The connection between the PANM and the identifiability of the causal direction is more concretely illustrated in Figure 2.1, showing cases when the true causal direction can and cannot be recovered. Depicted in example (a), edge $X - Y$ can be correctly oriented because the causal relations depicted in the DAG (top) yield the correctly specified pairwise ANM and the true parent sets $\text{pa}_{\mathcal{G}}(X) = \text{pa}_{\mathcal{G}}(Y) = \{A, B\}$, except the relation between X and Y, are identified in the PDAG (bottom). The independence relations hold for SEMs corresponding to $X \to Y$. Panel (b) shows an example in which an undirected edge satisfies the PANM even though the parent sets are not fully identified. The SEMs corresponding to $X \to Y$ are $Y = f_Y(X, B) + \varepsilon_Y$, which is the true SEM, and $X = \widetilde{\varepsilon}_X$, where $\widetilde{\varepsilon}_X = g(A, B) + \varepsilon_X$. While the parents

A, B for X are not detected in the PDAG as $\text{pa}_{\mathcal{G}}(X) = \varnothing$, they are merged with the error term ε_X to form the new error $\widetilde{\varepsilon}_X$. Thus, $X - Y$ satisfies the PANM and its orientation is identifiable.

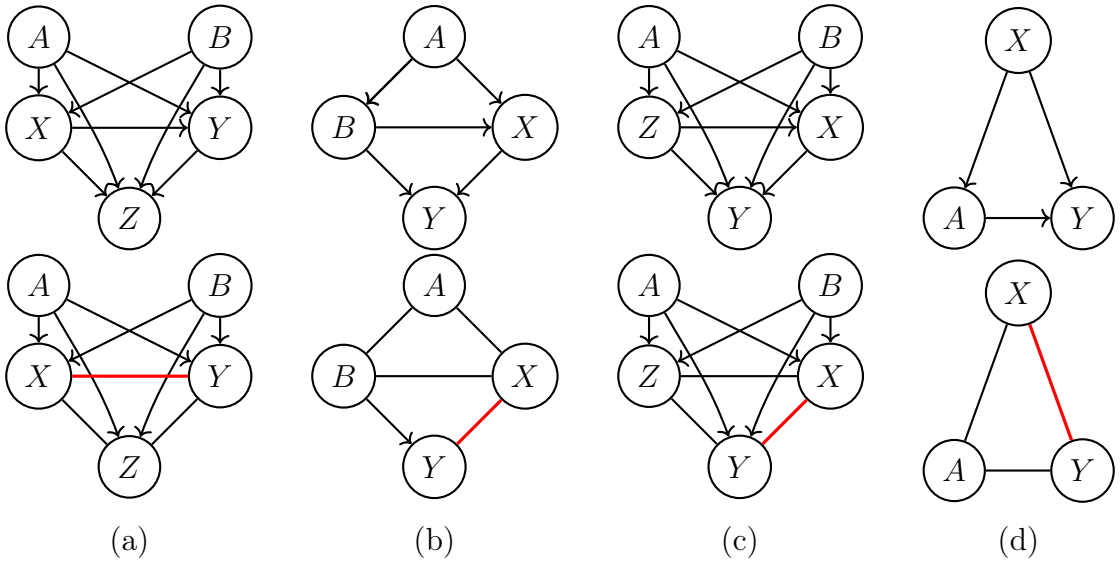


Figure 2.1: Examples to illustrate the PANM. The top row shows the true DAG, while the bottom row features a PDAG extendable to the true DAG with the evaluated edge $X - Y$. (a) $[X, Y \mid A, B]$ satisfies the PANM because both parent sets are fully identified. (b) $[X, Y \mid \varnothing, B]$ satisfies the PANM, despite A, B missing from $\text{pa}_{\mathcal{G}}(X) = \varnothing$, since we can write $X = \widetilde{\varepsilon}_X = g(A, B) + \varepsilon_X$. (c) $[X, Y \mid A, B]$ does not form a PANM, as common parent Z is not identified and becomes a latent confounder in the model. (d) $[X, Y]$ does not satisfy the PANM since node Y is missing parent A, which does not guarantee $\varepsilon_Y \perp\!\!\!\perp X$.

In example (c), however, the common parent node Z is not identified in the PDAG. The SEMs constructed over $[X, Y \mid A, B]$ would not satisfy either set of independence relations in Definition 2 due to the presence of a hidden confounder

Z, which would result in incorrect or inconclusive inference on the causal direction. In some cases, the independence relations entailed by the PANM may not hold if a parent node, say of Y, on a directed path from X to Y is not identified. Consider the undirected edge $X - Y$ in example (d), where $\text{pa}_{\mathcal{G}}(X) = \text{pa}_{\mathcal{G}}(Y) = \varnothing$ in the PDAG. Node A is a parent of Y in the DAG $\mathcal{G}_0$ on the directed path $X \to A \to Y$, but it is not identified as a parent of Y in the PDAG. We now examine whether the model $[X, Y]$ satisfies PANM under $X \to Y$. Node X has no parents in the true DAG, hence its SEM $X = \varepsilon_X$ matches the true form. Yet, node Y can only be expressed as $Y = f_Y(X, A) + \varepsilon_Y = f_Y(X, g(X) + \varepsilon_A) + \varepsilon_Y$, where A is substituted as $A = g(X) + \varepsilon_A$ and marginalized out. This shows that the SEM for Y is not an additive noise model. Suppose $\widehat{Y}$ is the best approximation of Y by functions of X assuming an additive noise. Then, the residual $Y - \widehat{Y}$ in general depends on X, so the independence relation $(Y - \widehat{Y}) \perp\!\!\!\perp X$ does not hold.

2.2.2 Key Idea: Sequential Edge Orientation

Central to our algorithm is a sequential edge orientation procedure. Given a PDAG $\mathcal{G}$, the procedure aims to determine the true causal direction of undirected edges in the graph. The core idea is to identify an undirected edge that satisfies the pairwise additive noise model, and then conduct a statistical test to determine its exact direction. We show that this procedure can be performed sequentially until all edges are oriented.

We present the high-level, population version of the sequential edge orientation procedure in Algorithm 1 to demonstrate this core idea. Given a PDAG, the algorithm identifies an edge (i, j) that satisfies the PANM on Line 2 and then orients the edge

into its true causal direction, which can be identified (Lemma 1). Subsequently, the algorithm leverages information read from $\mathcal{G}$ to further identify common children of i and j on Line 7, where we denote the set of neighbors and children of i as $\mathrm{nc}_{\mathcal{G}}(i) := \mathrm{ne}_{\mathcal{G}}(i) \cup \mathrm{ch}_{\mathcal{G}}(i)$. For each node $k \in \mathrm{nc}_{\mathcal{G}}(i) \cap \mathrm{nc}_{\mathcal{G}}(j)$, we orient the edges $i \to k$ and $j \to k$. Figure 2.2 features the three scenarios in which Line 7 is applicable: (1) $k \in \mathrm{ne}_{\mathcal{G}}(i) \cap \mathrm{ne}_{\mathcal{G}}(j)$, (2) $k \in \mathrm{ne}_{\mathcal{G}}(i) \cap \mathrm{ch}_{\mathcal{G}}(j)$, and (3) $k \in \mathrm{ch}_{\mathcal{G}}(i) \cap \mathrm{ne}_{\mathcal{G}}(j)$. Case 2 and case 3 respectively correspond to when $j \to k$ and $i \to k$ have been oriented by prior actions before evaluating $i - j$. For all three cases, we orient k as a common child of i and j as shown in the bottom panel of Figure 2.2. This is because of the following reasoning: If k were a common parent node of i, j, then $[i, j \mid \mathrm{pa}_{\mathcal{G}}(i), \mathrm{pa}_{\mathcal{G}}(j)]$ would not satisfy the independence relations entailed by the PANM since k would be a hidden confounder, as discussed in Figure 2.1c. If the true orientation were $i \to k \to j$ or $j \to k \to i$, this would be the case of Figure 2.1d and again would violate the PANM assumptions. Therefore, k must be a child node of both i and j. On the following line, the procedure applies rule 1 of Meek's orientation rules, where the configuration $a \to b - c$ is oriented as $a \to b \to c$ given there is no edge between a and c, to identify descendant nodes of a. Finally, if no undirected edges satisfy the condition on Line 2, we apply the Meek's rules to maximally orient the PDAG on Line 12.

Now we present a main result on Algorithm 1:

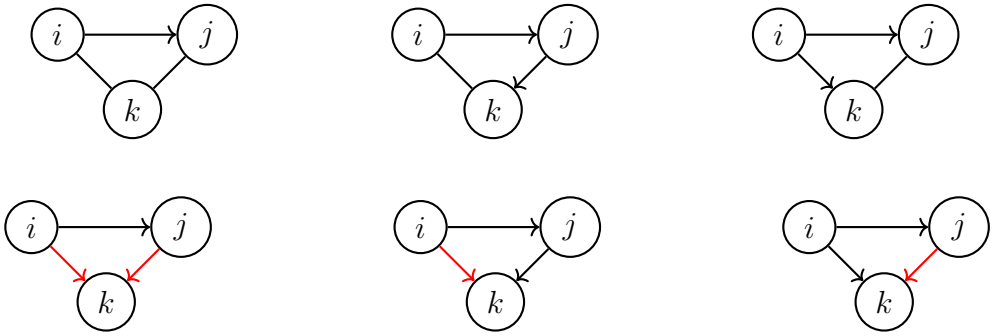
Theorem 1. *Suppose that $(X_1, \ldots, X_p)$ follows a restricted additive noise model with respect to a DAG $\mathcal{G}_0$. If the input $\mathcal{G}$ is the CPDAG of $\mathcal{G}_0$, then the sequential orientation procedure in Algorithm 1 orients $\mathcal{G}$ into the DAG $\mathcal{G}_0$.*

Theorem 1 shows that the edge orientation procedure in Algorithm 1 can recover

Algorithm 1: Sequential Orientation of Edges *(SequentialOrientation)*

Input: PDAG $\mathcal{G} = (V, E)$ and its undirected edges $U = E_U(\mathcal{G})$

while $|U| > 0$ **do**
  Search for $(i, j) \in U$ such that $[i, j \mid \mathrm{pa}_{\mathcal{G}}(i), \mathrm{pa}_{\mathcal{G}}(j)]$ satisfies the PANM;
  if *such* (i, j) *is found* **then**
    Identify the causal direction between i, j and orient (i, j) in $\mathcal{G}$ accordingly;
    (suppose $i \to j$ hereafter);
    if $nc_{\mathcal{G}}(i) \cap nc_{\mathcal{G}}(j) \neq \varnothing$ **then**
      Orient $i \to k$ and $j \to k$ in $\mathcal{G}$ for all $k \in \mathrm{nc}_{\mathcal{G}}(i) \cap \mathrm{nc}_{\mathcal{G}}(j)$;
    Apply Meek's orientation rule 1 to $\mathcal{G}$ repeatedly until it cannot be applied;
    Update $U \leftarrow E_U(\mathcal{G})$;
  else
    break
Apply all of Meek's rules to $\mathcal{G}$ repeatedly until none of them can be further applied.



(1) $k \in \mathrm{ne}_{\mathcal{G}}(i) \cap \mathrm{ne}_{\mathcal{G}}(j)$ (2) $k \in \mathrm{ne}_{\mathcal{G}}(i) \cap \mathrm{ch}_{\mathcal{G}}(j)$ (3) $k \in \mathrm{ch}_{\mathcal{G}}(i) \cap \mathrm{ne}_{\mathcal{G}}(j)$

Figure 2.2: Orientation rules of Line 7 in Algorithm 1. For each of the three cases in which $k \in \mathrm{nc}_{\mathcal{G}}(i) \cap \mathrm{nc}_{\mathcal{G}}(j)$ (top panel), we show the corresponding orientation of the red edge(s) in the bottom panel.

the true DAG from its CPDAG. A proof is provided in Appendix A.2.1. The key of the proof is to show that there always exists an undirected edge (i, j) in $\mathcal{G}$ that meets the condition in Line 2 as long as there are still undirected edges in $\mathcal{G}$. This is achieved by the careful design of the orientation rules from Line 4 to Line 8. To illustrate this point, suppose we did not include the orientation rule on Line 7 after orienting $i \to j$. As exemplified in case 1 of Figure 2.2, neither remaining undirected edges $i - k$ nor $j - k$ would satisfy the PANM. When evaluating $j - k$, node i is now a latent parent of k that would yield an error term dependent on X_j even under the true orientation $j \to k$. The case of $i - k$ corresponds to that in Figure 2.1d, which does not satisfy the PANM as we discussed. Therefore, the orientation rule on Line 7 not only identifies additional causal relations, but also ensures that there exists an undirected edge satisfying the PANM in the next iteration. Details on the existence of such an edge are expounded on in the proof.

In essence, our algorithm recovers the true DAG through two key steps: (1) to identify an edge (i, j) that satisfies the PANM model; (2) to infer the causal direction of (i, j) after it is identified. These key steps are achieved by our edge ranking and edge orientation procedures, which are introduced in Section 2.3. Precisely, we propose a criterion based on the pairwise additive noise model to identify an undirected edge for orientation and develop a likelihood-ratio test to infer its causal direction.

2.2.3 Algorithm Outline

The full SNOE algorithm is formally described in Algorithm 2, which implements the key idea of Algorithm 1 through three main steps: first to learn the initial CPDAG structure, then to orient the undirected edges in the CPDAG, and lastly to remove extraneous edges. In the most general case, the final output is a PDAG. However, practitioners may choose to output a DAG if they assume it is identifiable. See Remark 2 in Section 2.3.3 for details.

First, we apply a modified version of the PC algorithm [Spirtes and Glymour, 1991] to learn the initial structure (Lines 1–9). Specifically, we employ two significance levels: a stringent threshold α_1 to learn the CPDAG and a relaxed threshold α_2 to obtain a set of candidate edges. In our implementation, we use the partial correlation test to detect conditional independence relations. Starting with a complete, undirected graph, the PC algorithm removes edge (i, j) if nodes (X_i, X_j) are independent given a subset of their neighbors S_{ij}, tested at significance level α_1 (Lines 1–7). To obtain the candidate edges, we in fact first learn the skeleton using the relaxed significance level α_2 in the conditional independence tests, resulting in a denser skeleton as described on Lines 1–4. The candidate edges U_{α_2} (Line 10) are the edges removed when continuing

Algorithm 2: Causal Discovery by SNOE

Input: Observed data $X = (X_1, ..., X_p)$, complete undirected graph $\mathcal{G} = (V, E)$, sig. levels α_1, α_2, where $\alpha_2 > \alpha_1$

Output: PDAG $\mathcal{G} = (V, E)$

for $(i, j) \in E$ **do**
  Search for separating set $S_{ij} \subseteq V$ such that $p\text{-}val(X_i \perp\!\!\!\perp X_j | S_{ij}) > \alpha_2$;
  Update $E \leftarrow E \setminus \{(i, j), (j, i)\}$ and store S_{ij} if found;
$E_{\alpha_2} \leftarrow E$;
for $(i, j) \in E$ **do**
  Search for separating set $S_{ij} \subseteq V$ such that $p\text{-}val(X_i \perp\!\!\!\perp X_j | S_{ij}) > \alpha_1$;
  Update $E \leftarrow E \setminus \{(i, j), (j, i)\}$ and store S_{ij} if found;
Detect v-structures given E and $\{S_{ij}\}$;
Orient remaining undirected edges by Meek's rules;
Obtain candidate edge set $U_{\alpha_2} \leftarrow E_{\alpha_2} \setminus E$;
Merge edge sets to obtain $\mathcal{G} = (V, E \cup U_{\alpha_2})$;
Orient undirected edges in $\mathcal{G}$: $OrientEdges(X, \mathcal{G}, \alpha_1)$;
for $i = 1, \ldots, p$ **do**
  GAM regression $X_i \sim \{f_{i,k}(X_k) : k \in \mathrm{pa}_{\mathcal{G}}(i) \cup \mathrm{ne}_{\mathcal{G}}(i)\}$ and obtain $p\text{-}val(f_{i,k}(X_k))$ from significance testing for each k;
  if $p\text{-}val(f_{i,k}(X_k)) > \alpha_1$ **then**
    Update $E \leftarrow E \setminus \{(k, i)\}$;

the skeleton learning phase with α_1, and then are reintroduced to form the graph $\mathcal{G} = (V, E \cup U_{\alpha_2})$. The procedure is practically equivalent to learning a CPDAG under a strict significance level, then adding undirected edges between pairs of nodes with moderate association for consideration. We essentially separate skeleton learning and edge orientation into two tasks to obtain v-structures and directed edges with higher confidence, while preserving candidate edges to reduce the number of missing edges in the graphical structure. Although our work utilizes the PC algorithm, any causal discovery algorithm that learns the equivalence class of DAG $\mathcal{G}_0$, with multiple sparsity levels, would be compatible with our method.

The second stage aims to determine the true causal direction of undirected edges in the CPDAG. This is accomplished through our orientation procedure OrientEdges, which finds an evaluation order for undirected edges and then identifies their causal directions. To ensure that the undirected edges are correctly oriented in a sequential manner, we develop a measure to recursively rank undirected edges by their likelihood of satisfying the independence relations implied by the PANM. Then to orient an undirected edge $X - Y$, the edge orientation test LikelihoodTest, described in Algorithm 4, computes a likelihood ratio to compare the competing directions $X \to Y$ and $Y \to X$ given their learned parent sets $\mathrm{PA}_X, \mathrm{PA}_Y$ in the current PDAG $\mathcal{G}$. The test provides a definitive decision to either orient the edge in the preferred direction, if statistically significant, or leave it as undirected. The full details of the edge orientation procedure are presented in Algorithm 3.

In the third and last step, the algorithm removes extraneous edges in the graphical structure by covariate selection (Lines 13 – 16). Since the graph may contain undirected edges, the algorithm also considers neighbors when performing covariate selection. Recall that a neighbor of X is a node that shares an undirected edge with

X, excluding the parents and children of X in the graph. For a node X_i, the algorithm regresses X_i on its parents $\text{pa}_{\mathcal{G}}(X_i)$ and neighbors $\text{ne}_{\mathcal{G}}(X_i)$ using a generalized additive model (GAM). We perform significance testing and remove incoming edges from statistically insignificant nodes. For a neighbor $X_j \in \text{ne}_{\mathcal{G}}(X_i)$, the edge is oriented as $X_j \to X_i$ if $f_{i,j}(X_j)$ is statistically significant in the model for X_i and $f_{j,i}(X_i)$ is not significant in the model for X_j. If both terms are insignificant, the undirected edge is removed from the PDAG; otherwise, it remains intact.

Remark 1. In the implementation of Algorithm 2, we assume a causal additive model (1.7) for each node i. Accordingly, we use GAMs to complete all regression analysis in the algorithm. Since Theorem 1 applies to any identifiable additive noise model, one may replace GAM with other nonlinear regression techniques for a more general functional form of $f_i(\text{PA}_i)$.

2.3 Nonlinear Edge Orientation

To recover the true DAG from the learned CPDAG, we address two overarching questions: (1) how to determine the true causal direction of an undirected edge and (2) how to determine the evaluation order of edges. As discussed in Section 2.2.2, the core idea of SNOE is to identify and orient an undirected edge that, given the current parent sets in the PDAG, satisfies the pairwise additive noise model. In this section, we present how the two key components of SNOE, the edge ranking procedure and edge orientation test, resolve these challenges.

To determine the true causal direction, our method employs a likelihood ratio test, also referred to as the edge orientation test. Given an undirected edge $X - Y$ and the parent sets PA_X, PA_Y in the PDAG, the test compares the bivariate conditional

densities $p(X, Y \mid \mathrm{PA}_X, \mathrm{PA}_Y)$ factorized according to the directions $X \to Y$ and $Y \to X$. The likelihood ratio test can correctly identify the causal direction when $[X, Y \mid \mathrm{PA}_X, \mathrm{PA}_Y]$ satisfies the PANM. Furthermore, the test statistic exhibits a desirable asymptotic property that renders the result easy to obtain and interpret.

As previously shown in Figure 2.1, not every pair of nodes connected by an undirected edge satisfies the PANM. A violation of this assumption may cause incorrect conclusions in the orientation test. Thus, we develop an inference procedure to sort the undirected edges. We define a measure to quantify the adherence of an edge to the PANM, which is then utilized to determine edges eligible for orientation at a given stage. At every iteration in our sequential orientation procedure, there exists at least one edge following the PANM (Theorem 1), which is expected to be ranked and evaluated before all other undirected edges. By sequentially orienting the undirected edges in a correct order, the algorithm can ultimately learn the true DAG from the CPDAG.

2.3.1 Edge Orientation Algorithm

The full edge orientation procedure is presented in Algorithm 3. Lines 5 to 10 correspond to the procedure detailed in Algorithm 1. Before ordering the edges, the algorithm partitions undirected edges $\{U_k\}_{k=1}^m$ based on the number of neighbors $|\mathrm{ne}(U_k)|$ shared between the two nodes on Line 4. The undirected edges are then evaluated in subsets, starting with pairs of nodes sharing $|\mathrm{ne}(U_k)| = 0$ neighbors. Within each subset, the edges are ranked by an independence measure, which we utilize to determine their adherence to the PANM. This approach allows the algorithm to identify edges eligible for orientation more readily and reduce computation in

practice, as nodes with fewer shared neighbors are more likely to satisfy the PANM. We also apply all four of Meek's rules to further orient edges in $\mathcal{G}$, since the input PDAG may not be the true CPDAG in practice. This assists in reducing the number of undirected edges to sort.

Algorithm 3: Edge Orientation Procedure *(OrientEdges)*

Input: Observed data $X = \{X_i\}_{i=1}^p$, PDAG $\mathcal{G} = (V, E)$, sig. level α

Output: PDAG $\mathcal{G}$

Let $U = \{U_1, ..., U_m\}$ be the set of all undirected edges;
Calculate the number of common neighbors $|\text{ne}(U_k)|$ in $\mathcal{G}$ for each edge $U_k \in U$;
for $i = 0, \ldots, \max\{|\text{ne}(U_k)|\}$ **do**
    $\widetilde{U} \leftarrow \{U_k \in U : |\text{ne}(U_k)| = i\}$;
    Order edges in $\widetilde{U}$ by the edge-wise independence measure (2.2);
    for $j = 1, \ldots, |\widetilde{U}|$ **do**
        $\widetilde{U}_j = (a, b) \leftarrow$ *LikelihoodTest*$(\mathcal{G}, \widetilde{U}_j, X, \alpha)$;
        if $\widetilde{U}_j$ *is oriented* **then**
            Orient $a \to k$ and $b \to k, \forall k \in \text{nc}_{\mathcal{G}}(a) \cap \text{nc}_{\mathcal{G}}(b)$;
        Apply Meek's rules to $\mathcal{G}$ and update U accordingly;

Moreover, we may utilize the undirected components of a PDAG, defined below, to facilitate parallel orientation of undirected edges.

Definition 3 (Undirected Component in PDAG)**.** Let $\mathcal{G} = (V, E)$ be a PDAG, and $\mathcal{G}' = (V, E \setminus E_d)$ be the undirected graph obtained after removing all directed edges E_d of $\mathcal{G}$. We call a connected component of $\mathcal{G}'$ an *undirected component* of $\mathcal{G}$.

The undirected components provide practical significance in the edge orientation procedure. They not only isolate the set of undirected edges from directed edges, but also further partition the undirected edges into disjoint sets. Since the orientation of an undirected edge only affects the structure of its undirected component, edges in different undirected components can be evaluated and oriented in parallel. An efficient implementation is to apply Algorithm 3 separately to each undirected component of the input PDAG.

2.3.2 Ranking Undirected Edges by the PANM Criterion

As demonstrated in Lemma 1, the true orientation of an undirected edge $X - Y$ is identifiable when $[X, Y \mid \mathrm{pa}_{\mathcal{G}}(X), \mathrm{pa}_{\mathcal{G}}(Y)]$ follows the pairwise additive noise model. Given all the undirected edges, our ranking procedure positions such edges first for orientation by utilizing an independence measure derived from the independent noise property of the PANM.

Let us employ a pairwise dependence measure $I(X, Y)$ such that $I(X, Y) = 0$ if $X \perp\!\!\!\perp Y$ and $I(X, Y) > 0$ otherwise. Let $\widehat{X}$ and $\widehat{Y}$ denote the regression approximations of $\mathbb{E}[X \mid \mathrm{pa}_{\mathcal{G}}(X)]$ and $\mathbb{E}[Y \mid \mathrm{pa}_{\mathcal{G}}(Y), X]$ under $X \to Y$, where $\mathcal{G}$ is a PDAG as in Algorithm 3. Then for each undirected edge $X - Y$, we first calculate the maximum pairwise dependence between parents and residual of a node assuming the orientation $X \to Y$,

$$I(X \to Y) = \max_{Z,W} \{I(X - \widehat{X}, Z), I(Y - \widehat{Y}, W)\} \tag{2.1}$$

over all $Z \in \mathrm{pa}_{\mathcal{G}}(X)$ and $W \in \mathrm{pa}_{\mathcal{G}}(Y) \cup \{X\}$. For the opposite orientation, $I(Y \to X)$ is calculated similarly. The maximum pairwise dependence $I(X \to Y) = 0$ if the edge follows a PANM and the true orientation is $X \to Y$. Otherwise, we have

$I(X \to Y) > 0$. The edge-wise independence measure for $X - Y$, accounting for both possible directions, is the minimum of the two measures:

$$\widetilde{I}(X, Y) = \min[I(X \to Y), I(Y \to X)]. \tag{2.2}$$

Note that if edge $X - Y$ satisfies the PANM, then $\widetilde{I}(X, Y) = 0$. In our work, we use normalized mutual information as the pairwise dependence measure between two random variables Y_1 and Y_2,

$$I(Y_1, Y_2) = \frac{MI(Y_1, Y_2)}{\min[H(Y_1), H(Y_2)]}, \tag{2.3}$$

where $MI(\cdot, \cdot)$ is the mutual information and $H(\cdot)$ is the entropy measure. This dependence measure is bounded within $[0, 1]$ and more comparable across different pairs of random variables, as they may have quite different or extreme entropy measures. To simplify computation, we discretize all continuous variables for the calculation of mutual information and entropy. We employ sample splitting on the data to ensure the accuracy of this metric, where the data is split into training and test sets. To calculate the quantity $I(X - \widehat{X}, Z)$ in (2.1), for instance, we first fit a (nonlinear) regression model $\widehat{f}(\mathrm{PA}_X)$ for X using training data. Then we obtain the fitted value $\widehat{X} = \widehat{f}(\mathrm{PA}_X)$ from test data. Consequently, the residual $X - \widehat{X}$ and normalized mutual information $I(X - \widehat{X}, Z)$ are both calculated from test data, independent of training data, thus avoiding bias from model overfitting or reuse of the same data.

The purpose of this procedure is to distinguish edges that satisfy the PANM from those that do not. This is achieved simply by calculating $\widetilde{I}(\cdot, \cdot)$ for individual edges and sorting edges in ascending order of $\widetilde{I}(\cdot, \cdot)$. Naturally, this ranking produces an evaluation order for undirected edges, which is different from the common notion of a topological ordering of nodes.

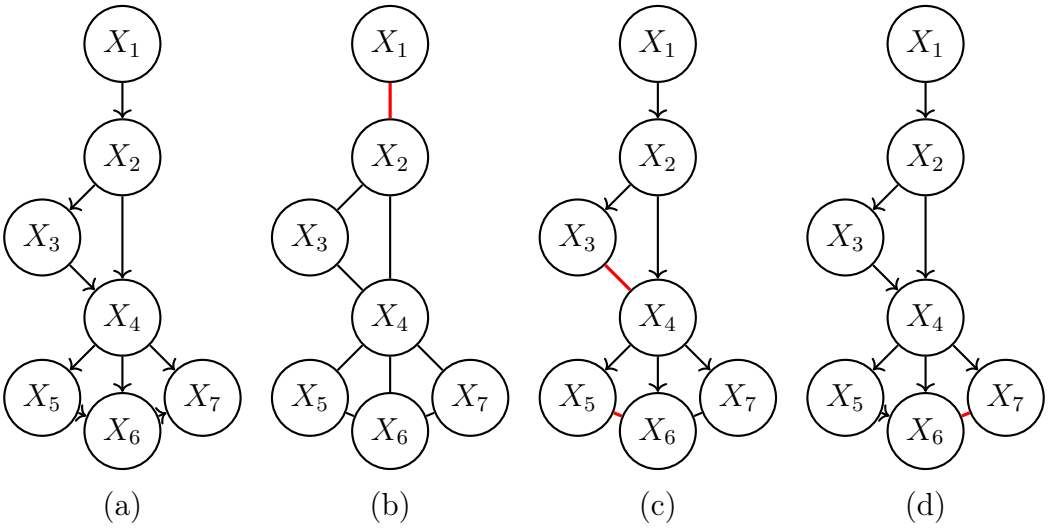


Figure 2.3: An illustration of the edge orientation procedure. (a) The true DAG. (b) The CPDAG, with $X_1 - X_2$ highlighted to orient first since it satisfies the pairwise ANM. (c) The resulting PDAG after orienting $X_1 \to X_2$ and employing Meek's rules. Edges $X_3 - X_4$ and $X_5 - X_6$ follow the pairwise ANM and can be oriented. (d) The true DAG is correctly recovered after orienting edge $X_6 - X_7$, which is ranked last due to missing the parent node X_5 for X_6 in (c).

The edge orientation procedure is illustrated through an example in Figure 2.3. In the CPDAG in Figure 2.3b, edges $\{X_1 - X_2, X_2 - X_3, X_4 - X_5\}$ follow the PANM and result in $\widetilde{I}(\cdot, \cdot) = 0$. Yet since Algorithm 3 considers pairs of nodes sharing no neighbors first, it first orients $X_1 - X_2$. As a result of orienting $X_1 \to X_2$ and applying Meek's rules, we obtain the maximally oriented PDAG shown in Figure 2.3c with several more directed edges uncovered. Two undirected components, $\{X_3, X_4\}$ and $\{X_5, X_6, X_7\}$, of the PDAG can now be oriented in parallel. The algorithm would then find $\widetilde{I}(X_3, X_4) = 0$ and $\widetilde{I}(X_5, X_6) = 0$ because both edges satisfy the PANM, and $\widetilde{I}(X_6, X_7) > 0$ due to X_5 missing from $\text{pa}_{\mathcal{G}}(X_6) = \{X_4\}$. Therefore, our method would rank and evaluate $X_3 - X_4$ and $X_5 - X_6$ before $X_6 - X_7$. After applying the orientation test and rules again, the algorithm would orient the last undirected edge $X_6 - X_7$ to recover the true DAG, as seen in Figure 2.3d.

2.3.3 Likelihood Ratio Test for Edge Orientation

Our method adopts a comprehensive approach to edge orientation by considering the subgraph formed by both nodes and their learned parent sets. The likelihood ratio test returns a clear decision for edge orientation, whereas the causal relation is difficult to interpret when separate independence tests for opposite edge directions both return statistically significant outcomes [Shah and Peters, 2020]. It is also more robust against violations of the model assumptions, e.g. the noise distribution. Given that an undirected edge meets the PANM criterion, the algorithm applies the test to determine its causal direction. We first introduce the formulation of the test statistic and then describe the testing procedure.

Our orientation test takes inspiration from Vuong's test, a series of likelihood ratio

tests for model selection and testing non-nested hypotheses [Vuong, 1989]. Consider two nodes X, Y connected by an undirected edge in a PDAG, where $Z_1 = \mathrm{PA}_X$ and $Z_2 = \mathrm{PA}_Y$ have been identified. As indicated in Lemma 1, the PANM is only fulfilled under one causal direction. If the true direction is $X \to Y$, a reverse causal model $Y \to X$ would not satisfy the independence relations in Definition 2 nor adequately fit to the joint distribution. Building on this insight, we compare two sets of conditional models that factorize the joint conditional density $p(x, y \mid z_1, z_2)$ according to the opposing directions:

$$\text{Under } X \to Y : F_{\theta^*}(x, y \mid z_1, z_2) = p(y \mid z_2, x; \theta_1^*)p(x \mid z_1; \theta_2^*), \tag{2.4}$$

$$\text{Under } Y \to X : G_{\gamma^*}(x, y \mid z_1, z_2) = p(x \mid z_1, y; \gamma_1^*)p(y \mid z_2; \gamma_2^*). \tag{2.5}$$

The two conditional densities are parameterized respectively by $\theta^* = (\theta_1^*, \theta_2^*)$ and $\gamma^* = (\gamma_1^*, \gamma_2^*)$. We perform two-fold sample splitting on the observed data set, where the training data is used to estimate model parameters $(\widehat{\theta}, \widehat{\gamma})$ and the test data is used to evaluate the log-likelihood. This ensures that $(\widehat{\theta}, \widehat{\gamma})$ are independent of the test data (X_i, Y_i) and allows us to establish the asymptotic distribution of the log-likelihood ratio.

To determine the edge orientation, we consider three hypotheses in the likelihood-ratio test. The null hypothesis is given as

$$H_0 : \mathbb{E}\left[\log \frac{F_{\theta^*}(X, Y \mid Z_1, Z_2)}{G_{\gamma^*}(X, Y \mid Z_1, Z_2)}\right] = \mathbb{E}\left[\log \frac{p(Y \mid Z_2, X; \theta_1^*)p(X \mid Z_1; \theta_2^*)}{p(X \mid Z_1, Y; \gamma_1^*)p(Y \mid Z_2; \gamma_2^*)}\right] = 0 \tag{2.6}$$

and the two alternative hypotheses are formulated as

$$H_f : \mathbb{E}\left[\log \frac{F_{\theta^*}(X, Y \mid Z_1, Z_2)}{G_{\gamma^*}(X, Y \mid Z_1, Z_2)}\right] > 0 \quad \text{and} \quad H_g : \mathbb{E}\left[\log \frac{F_{\theta^*}(X, Y \mid Z_1, Z_2)}{G_{\gamma^*}(X, Y \mid Z_1, Z_2)}\right] < 0. \tag{2.7}$$

The test uses the likelihood ratio statistic to select the model closest to the true conditional distribution. The null hypothesis H_0 indicates that the likelihood is comparable between the two candidate models, hence we cannot identify the causal direction from the observed data. The alternative hypotheses, H_f and H_g, are accepted when a particular edge direction is more probable. The variance of the log-likelihood ratio with respect to the joint distribution $[X, Y, Z_1, Z_2]$ is denoted as

$$\omega_*^2 = \mathbb{V}\mathrm{ar}\left[\log \frac{F_{\theta^*}(X, Y \mid Z_1, Z_2)}{G_{\gamma^*}(X, Y \mid Z_1, Z_2)}\right]. \tag{2.8}$$

When the conditional models are equivalent, i.e. $F_{\theta^*} = G_{\gamma^*}$, we have $\omega_*^2 = 0$.

We first establish a general result for the log-likelihood ratio with sample splitting. Let (X, Z) denote a generic observation. Suppose two models $F(\cdot \mid Z)$ and $G(\cdot \mid Z)$ are estimated using an independent training sample of size n, yielding estimators $\widehat{F}(\cdot \mid Z)$ and $\widehat{G}(\cdot \mid Z)$. Let $\{(X_i, Z_i)\}_{i=1}^n$ be an independent test sample.

Define the test-sample log-likelihood ratio

$$R_i = \log \frac{\widehat{F}(X_i \mid Z_i)}{\widehat{G}(X_i \mid Z_i)}, \qquad i = 1, \ldots, n,$$

and let

$$\bar{R} = \frac{1}{n}\sum_{i=1}^n R_i, \qquad s_R^2 = \frac{1}{n-1}\sum_{i=1}^n (R_i - \bar{R})^2.$$

The population-level parameters are

$$\mu_0 = \mathbb{E}\left[\log \frac{F(X \mid Z)}{G(X \mid Z)}\right], \qquad \sigma_0^2 = \mathbb{V}\mathrm{ar}\left[\log \frac{F(X \mid Z)}{G(X \mid Z)}\right].$$

Let $D(p \,\|\, q) := \mathbb{E}_{(X,Z)}[\log(p(X \mid Z)/q(X \mid Z))]$ for two conditional densities $p(x \mid z)$ and $q(x \mid z)$, where the expectation is taken with respect to the true joint distribution of (X, Z). If p is the true distribution, then $D(p \,\|\, q) = \mathbb{E}_Z\left[\mathrm{KL}(p(\cdot \mid Z) \,\|\, q(\cdot \mid Z))\right]$, where KL denotes the Kullback–Leibler (KL) divergence.

Theorem 2. *Suppose the estimators $\widehat{F}$ and $\widehat{G}$ are measurable with respect to the training sample of size n and independent of the test sample $\{(X_i, Z_i)\}_{i=1}^n$. Assume the following conditions:*

(i) There exists $\eta > 0$ such that

$$\sup_{n \geq 1} \mathbb{E}\left[\left|\log \frac{\widehat{F}(X \mid Z)}{\widehat{G}(X \mid Z)}\right|^{2+\eta}\right] < \infty.$$

(ii) As $n \to \infty$, the conditional variance

$$v_n = \mathbb{V}\mathrm{ar}\left[\log \frac{\widehat{F}(X \mid Z)}{\widehat{G}(X \mid Z)} \;\middle|\; \widehat{F}, \widehat{G}\right] \xrightarrow{p} \sigma_0^2, \qquad 0 < \sigma_0^2 < \infty.$$

(iii) With respect to the distribution of the training sample,

$$D\big(F \,\|\, \widehat{F}\big) = o_p(n^{-b}), \qquad D\big(G \,\|\, \widehat{G}\big) = o_p(n^{-b}), \qquad b \geq 0.$$

Then, as $n \to \infty$, $\bar{R} \xrightarrow{p} \mu_0$ if $b = 0$ and

$$\frac{\sqrt{n}\,(\bar{R} - \mu_0)}{s_R} \xrightarrow{D} N(0, 1)$$

if $b = 1/2$.

Theorem 2 shows that the log-likelihood ratio converges to the standard normal distribution in the large sample limit, under assumptions (i) to (iii) if the convergence rates in (iii) are $o(n^{-1/2})$. Assumption (i) requires the $(2 + \eta)$-th moment of the log-likelihood ratio to be finite such that $\bar{R}$ and s_R^2 are well-behaved and converge to their respective true values. Under (ii), the conditional variance stabilizes across training samples. Assumption (iii) states that the loss of the conditional density estimators, measured by $D(\cdot\|\cdot)$, converges at rate $o(n^{-1/2})$. This rate can be achieved

by many common nonparametric methods under certain smoothness assumption for the densities. Moreover, the rate can be relaxed to $o(1)$ with $b = 0$ if we only need to establish convergence in probability for $\bar{R}$, which is relevant for H_f and H_g. Note that neither F nor G may be the true conditional distribution of X given Z. They are usually defined as the minimizer of the expectation of an empirical loss.

Given samples $\{X_i, Y_i, Z_{1,i}, Z_{2,i}\}_{i=1}^n$, we define the likelihood ratio statistic and the estimated variance as

$$LR_n(\widehat{F}, \widehat{G}) = \sum_{i=1}^{n} \log \frac{\widehat{F}(X_i, Y_i \mid Z_{1,i}, Z_{2,i})}{\widehat{G}(X_i, Y_i \mid Z_{1,i}, Z_{2,i})}, \tag{2.9}$$

$$\widehat{\omega}_n^2 = \mathbb{V}\mathrm{ar}\left\{ \log \frac{\widehat{F}(X_i, Y_i \mid Z_{1,i}, Z_{2,i})}{\widehat{G}(X_i, Y_i \mid Z_{1,i}, Z_{2,i})} \;\middle|\; \widehat{F}, \widehat{G} \right\}_{i=1}^{n}. \tag{2.10}$$

We now establish the following asymptotic result for the likelihood ratio test, as an immediate consequence of Theorem 2.

Proposition 1. *Suppose the estimators $\widehat{F}, \widehat{G}$ are independent of $\{(X_i, Y_i, Z_{1,i}, Z_{2,i}), i \in [n]\}$ and satisfy conditions (i) and (ii) in Theorem 2. Let $\omega_*^2 > 0$. If H_0 (2.6) is true and condition (iii) holds for $b = 1/2$, then*

$$\frac{LR_n(\widehat{F}, \widehat{G})}{\sqrt{n}\widehat{\omega}_n} \xrightarrow{D} N(0, 1), \qquad n \to \infty. \tag{2.11}$$

If H_f is true and condition (iii) holds for $b = 0$, then

$$\frac{LR_n(\widehat{F}, \widehat{G})}{\sqrt{n}\widehat{\omega}_n} \xrightarrow{p} +\infty, \qquad n \to \infty. \tag{2.12}$$

By symmetry, the above test statistic converges to $-\infty$ under H_g. The test is applicable to $\widehat{F}$ and $\widehat{G}$ obtained by either parametric or nonparametric methods. To

interpret the final statistic, the standard decision rule in hypothesis testing applies according to the normal distribution in (2.11). For a chosen significance level, if the null hypothesis is not rejected, we leave the edge as undirected. Otherwise, the test has detected a probable causal direction and the edge is oriented accordingly. In contrast to kernel-based tests, the asymptotic property of our likelihood ratio test statistic makes the orientation test computationally tractable, thereby enabling efficient and reliable inference of the causal direction. This test also exhibits an advantage over a score-based approach. Rather than choosing the edge direction with a higher likelihood value to improve the score, the p-value quantifies the statistical significance and uncertainty for the magnitude of the likelihood ratio.

An outline of the edge orientation test is given in Algorithm 4. When the conditional models F_{θ^*} and G_{γ^*} are equivalent, conducting the likelihood ratio test is unnecessary as the edge direction is practically indistinguishable. To test for model equivalence, we devise a variance test to assess whether $\omega_*^2 = 0$, as seen on Line 3. The test compares the sample variance $\widehat{\omega}_n^2$ to

$$v^2 \triangleq \min\left\{\mathbb{V}\mathrm{ar}(\log \widehat{F}_n(X, Y \mid \mathrm{PA}_X, \mathrm{PA}_Y)), \mathbb{V}\mathrm{ar}(\log \widehat{G}_n(X, Y \mid \mathrm{PA}_X, \mathrm{PA}_Y))\right\},$$

which is the smaller variance of the log-likelihood estimates computed under one direction. If $\widehat{\omega}_n^2/v^2 < \delta$, for some small threshold δ, then we bypass the likelihood test and declare the edge direction as indistinguishable.

Remark 2. While the final output of Algorithm 2 is a PDAG, users may specify to return a DAG if they believe that the nonlinear ANM assumption holds. Then, in an additional fourth and final stage, the edge orientation procedure in Algorithm 3 is applied again to extend the PDAG to a DAG. If any undirected edge still remains, the algorithm chooses the orientation with a higher log-likelihood value as the inferred

Algorithm 4: Test for Edge Orientation *(LikelihoodTest)*

Input: PDAG $\mathcal{G}$, undirected edge $X - Y$, observed data

$\{X, Y, \mathrm{PA}_{\mathcal{G}}(X), \mathrm{PA}_{\mathcal{G}}(Y)\}$, sig. level α

Output: Edge (X,Y) $\in \{X \to Y, Y \to X, X - Y\}$

Perform train-test split on observed data;

Estimate models $\widehat{F}$ and $\widehat{G}$ using GAMs with training data;

Conduct variance test on $\widehat{\omega}_n^2$ (2.10) for model equivalence;

if $\widehat{\omega}_n^2 / v^2 > \delta$ **then**

  Compute the likelihood ratio test statistic $LR_n(\widehat{F}, \widehat{G})$ (2.9);

  Obtain p-value $p_{E_{X,Y}}$ and preferred edge direction $E_{X,Y}$:

$$E_{X,Y} = \begin{cases} X \to Y, \text{ if } LR_n(\widehat{F}, \widehat{G}) > 0 \\ Y \to X, \text{ otherwise} \end{cases} .$$

  if $p_{E_{X,Y}} < \alpha$ **then**

    Orient edge (X, Y) as $E_{X,Y}$.

causal direction of the edge.

Remark 3. Our full algorithm involves constructing regression models in several tasks. This includes estimating residuals for computing $\widetilde{I}(X, Y)$ to rank edges, fitting models to compute the log-likelihood values under possible configurations in subgraphs, and performing covariate testing in the last stage. In the software implementation, the algorithm utilizes generalized additive models from the **mgcv** package [Wood, 2001] to construct regression models, with the thin plate spline selected as the basis function.

2.4 Structural Learning Consistency

In this section, we establish the correctness of our algorithm in the large-sample limit based on the validity of the sequential orientation procedure at the population level stated in Theorem 1. There are two key elements for demonstrating the consistency of the algorithm. The first key element is to establish the consistency of the nonlinear regression methods. The second is to establish the consistency of the tests utilized in various steps in our algorithm, namely the initial CPDAG learning stage, the ranking of undirected edges for evaluation, and the orientation of undirected edges.

Remark 4. To ease the exposition of technical details, we consider a simplified version of Algorithm 2, in which we do not partition undirected edges based on the number of neighbors in Algorithm 3. Instead, we simply rank all undirected edges by the independence measure $\widetilde{I}$ after each round of edge orientation. We only consider the initial learning and edge orientation phases, as the edge pruning phase is not needed in the large-sample limit. Moreover, we apply Meek's rules according to Algorithm 1. Our consistency results in this section are established for this simplified Algorithm 2.

First, we define population regression functions and the associated residual variables. The population regression function for X_i given subset $S \subseteq [p] \setminus \{i\}$ and the associated residual variable are defined as

$$g_{i,S} := \underset{h \in \mathcal{F}^{\oplus k}}{\arg\min}\ \mathbb{E}\left[X_i - h(X_S)\right]^2, \tag{2.13}$$

$$\varepsilon_{i,S} := X_i - g_{i,S}(X_S), \tag{2.14}$$

where $k = |S|$ and $\mathcal{F}^{\oplus k}$ is the space of additive functions defined in (1.8). Let $\sigma^2_{i,S} := \mathbb{V}\mathrm{ar}(\varepsilon_{i,S}) > 0$ be the variance of $\varepsilon_{i,S}$. We make the following assumptions on the error variables and the estimation of $g_{i,S}$ and $\sigma^2_{i,S}$.

Assumption 1. For all $i \in [p]$ and $S \subseteq [p] \setminus \{i\}$, $\mathbb{E}|\varepsilon_{i,S}|^{4+2\eta} < \infty$ for some $\eta > 0$ and the estimators $(\widehat{g}_{i,S}, \widehat{\sigma}^2_{i,S})$ constructed with a sample of size n satisfy:

$$c \le \widehat{\sigma}^2_{i,S} \le M \text{ for all } n, \tag{2.15}$$

$$\sup_n \mathbb{E}\Big[\big|\widehat{g}_{i,S}(X_S)) - g_{i,S}(X_S)\big|^{4+2\eta}\Big] < \infty. \tag{2.16}$$

$$\mathbb{E}[\widehat{g}_{i,S}(X_S) - g_{i,S}(X_S)]^2 = o(n^{-1/2}), \tag{2.17}$$

$$\mathbb{E}(\widehat{\sigma}^2_{i,S} - \sigma^2_{i,S})^2 = o(n^{-1/2}), \tag{2.18}$$

where $X_S \perp\!\!\!\perp (\widehat{g}_{i,S}, \widehat{\sigma}^2_{i,S})$ and $0 < c < M < \infty$ are constants.

Assumptions (2.15) and (2.16) are mild, only requiring boundedness of $\widehat{\sigma}^2_{i,S}$ and a finite moment of the MSE of $\widehat{g}_{i,S}$. The L_2 convergence rates in (2.17) and (2.18) are the key assumptions, which imply Condition (iii) of Theorem 2 for Gaussian additive regression; see Appendix A.1 for more details. If each additive function in $g_{i,S}$ belongs to the Hölder class of smoothness q, the classical rate for the MSE of $\widehat{g}_{i,S}$ is $n^{-2q/(2q+1)}$ [Stone, 1985], which is faster than $n^{-1/2}$ for all $q > 1/2$. The standard parametric rate $n^{-1/2}$ of $\widehat{\sigma}^2_{i,S}$ gives an MSE of $O(n^{-1})$.

We now list a few additional assumptions and formally state the consistency of the algorithm.

Assumption 2. Suppose $\{X_i\}_{i=1}^p$ follows a CAM (1.7) with a faithful DAG $\mathcal{G}_0$ and satisfies the following assumptions:

(A1) For any $i, j \in [p]$ and any $S \subset [p]$, if $X_i \not\perp\!\!\!\perp X_j \mid X_S$, then the partial correlation $|\rho_{ij|S}| > \tau$ for some $\tau > 0$.

(A2) For any $i \in [p]$, if $S \subseteq \mathrm{pa}_{\mathcal{G}_0}(i) \cup \mathrm{ch}_{\mathcal{G}_0}(i)$ and $S \cap \mathrm{ch}_{\mathcal{G}_0}(i) \neq \varnothing$, then the mutual information $MI(\varepsilon_{i,S}, X_k) > \delta$ for some $k \in S$ and some constant $\delta > 0$.

(A3) For any $i \in [p]$ and any $S \subseteq \mathrm{pa}_{\mathcal{G}_0}(i) \cup \mathrm{ch}_{\mathcal{G}_0}(i)$, the entropy measures $H(X_i), H(\varepsilon_{i,S}) \in [c_1, c_2]$, where $c_2 > c_1 > 0$ are constants.

In (A2) and (A3), mutual information and entropy measures are calculated through discretization with positive cell probability on every bin.

Theorem 3. *Let $\widehat{\mathcal{G}}_n$ be the learned graph of the simplified Algorithm 2 applied to an i.i.d. sample of size n, in which all involved regression problems are estimated with $(\widehat{g}_{i,S}, \widehat{\sigma}^2_{i,S})$ via Gaussian regression. If Assumptions 1 and 2 hold, then*

$$\lim_{n\to\infty} \mathbb{P}(\widehat{\mathcal{G}}_n = \mathcal{G}_0) \to 1 \tag{2.19}$$

for some choice of $\alpha_1, \alpha_2 \to 0$.

This result establishes the consistency of the algorithm in learning the true DAG. Assumption 1 is sufficient for the consistency of computing the normalized mutual information using estimated residuals and that of the likelihood ratio test for inferring the causal direction of undirected edges. The consistency of $\widehat{\mathcal{G}}_n$ also relies on the

consistency of the statistical tests performed. Pertinent to skeleton learning in stage 1, Assumption 2 (A1) states that there exists a lower bound $\tau > 0$ for the partial correlation when $X_i \not\perp\!\!\!\perp X_j \mid S_{ij}$. We show that the probabilities of type I and type II errors converge to 0 for the CI tests in the large sample limit, by which our algorithm obtains a consistent CPDAG in stage 1. Assumptions 2 (A2) and (A3) are pertinent to identifying undirected edges that satisfy the PANM criterion. We assume the existence of a gap $\delta > 0$ for $MI(\varepsilon_{i,S}, X_k), k \in S$ to precisely distinguish edges that do and do not follow the PANM. Last, we assume a mild boundedness assumption on the entropy of each X_i and various residual variables, which guarantees that the normalized independence measure (2.3) is well-defined.

Remark 5. The proof of Theorem 3 does not rely on the additive function assumption in (1.7) except for its identifiability. Thus, the structure learning consistency of our algorithm can be readily generalized to the larger class of identifiable ANMs with Gaussian noise.

We analyze the computational complexity by counting the number of statistical tests performed and regression models fitted in the algorithm. For a $p-$node problem, the learned CPDAG can be a complete graph consisting of $p(p-1)/2$ edges in the worst-case scenario. The CPDAG $\mathcal{E} = (V, E)$ generally contains much fewer edges and its undirected edges $\{U\}_{i=1}^{m}, m < |E|$ generally account for only a fraction of all edges. To compare two causal directions in the edge orientation procedure, our method builds two models for each direction and conducts one test per direction. The edge ranking procedure performs $2m$ tests and fits $4m$ regression models, while the orientation procedure performs at most $|U| = m$ tests and fits $4m$ models. However, there are fewer tests and models required in practice because Meek's rules will orient

additional edges. The computational complexity of procedure *OrientEdges* is then of order $O(m)$. In the edge deletion step, the method performs one significance test per node on its covariates, amounting to p tests and p models, and has a complexity of order $O(p)$. While the PC algorithm only conducts conditional independence tests and is exponential with respect to p in the worst case, it becomes polynomial when the underlying DAG is sparse.

We present the performance of our algorithm under a series of experiments in Chapter 3.

2.5 Discussion

In this work, we present a novel algorithm for learning nonlinear causal DAGs through a sequential edge orientation framework. Specifically, we demonstrate that the edge orientation algorithm can learn the true DAG from the CPDAG by sequentially orienting the undirected edges. The framework is established on the pairwise additive noise model, a criterion we developed to ensure accurate inference of the causal direction for undirected edges from just the two nodes and their identified parent sets. The sequential orientation of edges is achieved through two key components: the likelihood ratio test, which provides a definitive decision on the causal direction of an undirected edge, and the edge ranking procedure, which recursively determines edges that follow the PANM to ensure the correctness of orientations made. These two procedures effectively address two fundamental questions for constraint-based approaches regarding how to determine the causal direction of edges and how to order edges for evaluation. We also propose two approaches to the likelihood ratio test, both of which demonstrate well-controlled type I error and high statistical

power. SNOE provides a practical, yet still precise, alternative to kernel-based and regression-based learning of nonlinear causal relations. Compared to competing methods, SNOE exhibits robustness and high precision, which can be attributed to its reduced dependence on model assumptions. It also requires far less computational time and demonstrates strong generalizability to different data functions and distributions.

Potential extensions of this framework include learning nonlinear causal relations in the presence of hidden variables — a common challenge in constraint-based algorithms. The key is to adapt the edge ranking and the likelihood ratio test to take into account latent confounders. In addition, this work can be refined and expanded in several ways. One direction is to expand our algorithm to the post-nonlinear (PNL) model, where the SEM is given as $X_i = g_i(f_i(\mathrm{PA}_i) + \varepsilon_i)$ and the causal direction can be determined by testing the independence between ε_i and PA_i [Zhang and Hyvärinen, 2009]. We can adapt the pairwise ANM criterion to the PNL model, as it also relies on the independence noise property for identification, and thereby produce an evaluation order of edges in learning the true DAG from its MEC. Deep learning methods [Uemura and Shimizu, 2020] have been developed to better approximate f_i, g_i and the noise term, which would allow us to accurately compute the normalized MI, rank edges, and perform the likelihood ratio test under the PNL model. Another promising direction is to integrate alternative independence measures and estimation methods into our algorithm. For general nonlinear models, a kernel-based test [Zhang et al., 2012] may be more accurate and an alternative likelihood estimation, such as the method proposed in Khemakhem et al. [2021], can be used in place of the likelihood ratio test to determine the causal direction.

CHAPTER 3

Numerical Results

We conducted numerical experiments with synthetic data to benchmark the accuracy and effectiveness of our method. At a detailed level, we assess the performance of the ranking procedure in Section 3.1.1, as well as the type I error rate and statistical power of the likelihood ratio test in Section 3.1.2. We then compare our method to competing causal discovery algorithms using simulated data sets in Sections 3.1.3. Intermediate results from each stage of our algorithm are provided in Section 3.1.4 to illustrate the effects of the individual components. Two real-world applications are presented in Section 3.2.

3.1 Numerical Experiments on Synthetic Data

We develop two variations of the likelihood ratio test in our algorithm: the *sample-splitting (SNOE-SS)* approach, which is delineated in Algorithm 4, and the *cross-validation (SNOE-CV)* approach. The CV approach employs two-fold cross-validation in Algorithm 4 to perform the likelihood ratio test twice by exchanging the training and test data sets and uses either the smaller or larger p-value for evaluation. The larger p-value is used in our experiments, but practitioners may specify either option. To learn the initial graph, we implemented our modified version of the PC-stable

algorithm [Colombo and Maathuis, 2014] from the **bnlearn** package coupled with the partial correlation test [Scutari, 2010]. A more stringent threshold of $\alpha_1 = 0.05$ was applied for learning the CPDAG, while a relaxed threshold of $\alpha_2 = 0.25$ was used for obtaining the additional candidate edges. The significance level for the likelihood ratio test was set at $\alpha = 0.05$ and further reduced to $\alpha = 10^{-4}$ for edge pruning. The algorithm is implemented as an R package and can be accessed at `https://github.com/stehuang/snoe.git`.

3.1.1 Accuracy of Ranking Procedure

We performed several experiments to verify the precision of our edge ranking procedure. Specifically, we tested its ability to correctly rank undirected edges in a CPDAG. Three distinct DAG structures were considered, as depicted in Figure 3.1, with $N = 200$ data sets generated for each structure, each containing $n = 2000$ samples. As the data was generated using nonlinear functions, the edge directions can be determined under all settings. Given the CPDAG of each DAG, we computed the edge-wise independence measure $\widetilde{I}(X, Y)$, defined in (2.2), for each undirected edge $X - Y$ and ranked the edges in ascending order. We assessed whether an edge satisfying the PANM was ranked first.

Results are presented in Table 3.1 and are categorized by the data generating function used in the simulations. Note that the cubic, piecewise linear, and sigmoid functions are invertible. The values represent the proportion of data sets in which an undirected edge was ranked first for orientation. Uniformly across all graphs and functions, the edge following the PANM has the highest proportion of being ranked first. It is identified in the vast majority of data sets generated by the quadratic

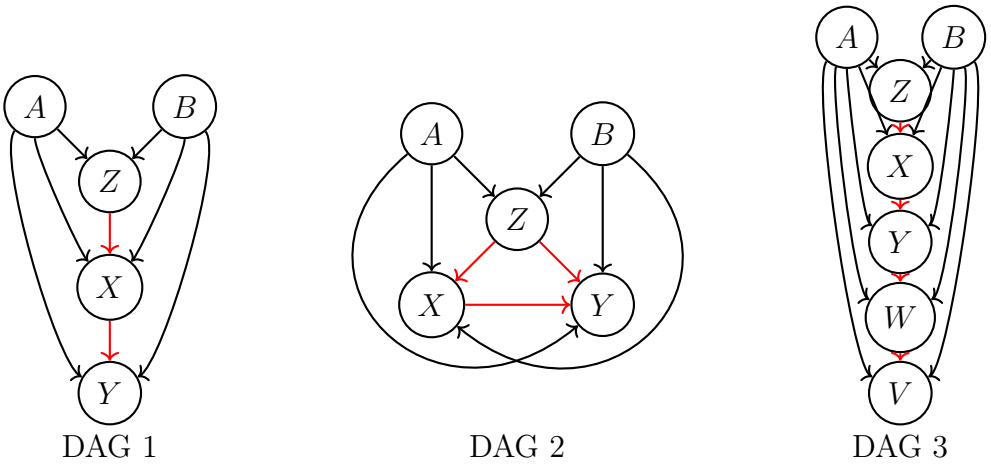


Figure 3.1: Example graphs used for testing the ranking procedure. Red edges indicate the undirected edges in the CPDAG that are evaluated and ranked by the procedure.

and sigmoid functions. Notably, our ranking procedure successfully determined the edge that satisfies the PANM in most cases under the cubic and piecewise functions, which is more challenging to distinguish since these invertible functions can be well-approximated by a linear function. These experiments verify both the ranking procedure and the use of the edge-wise independence measure to identify an edge fulfilling the orientation criterion.

3.1.2 Evaluation of the Likelihood Ratio Test

In this subsection, we investigate the type I error and statistical power of the likelihood ratio test. We considered five distinct DAG structures, each comprising of 2 to 5 nodes and 1 to 7 edges, and generated data sets of sample sizes $n = 250, 500, 1000, 1500, 2000$. In the CPDAG of each network, we applied the likelihood ratio test to a targeted

DAG Structure	Edge	Satisfies PANM	Cubic	Piecewise	Quadratic	Sigmoid
1	$Z - X$	Yes	0.64	0.62	0.99	0.72
1	$X - Y$	No	0.36	0.38	0.01	0.28
2	$Z - X$	Yes	0.43	0.52	0.46	0.93
2	$Z - Y$	No	0.34	0.25	0.35	0.07
2	$X - Y$	No	0.23	0.23	0.19	0
3	$Z - X$	Yes	0.38	0.52	0.95	0.41
3	$X - Y$	No	0.14	0.24	0.01	0.19
3	$Y - W$	No	0.26	0.14	0.01	0.21
3	$W - V$	No	0.22	0.11	0.03	0.19

Table 3.1: Frequency of an undirected edge ranked first under various settings.

undirected edge to determine its causal direction. A total of $N = 400$ tests were performed per graphical structure and sample size setting. Under a linear Gaussian DAG, the true edge direction of the targeted undirected edge is not identifiable. Under a nonlinear DAG, the true edge direction is identifiable.

A type I error under the likelihood ratio test would be to declare one model more probable than the other when the two models are equivalent. In the context of structural learning, this occurs when an undirected edge is oriented, but should remain undirected. To quantify the type I error rate, we applied the likelihood ratio test to an undirected edge in the CPDAG of a linear, Gaussian DAG and recorded the errors made under significance levels $\alpha = 0.01, 0.05$.

The type I error of the test, averaged across all DAG structures per sample size,

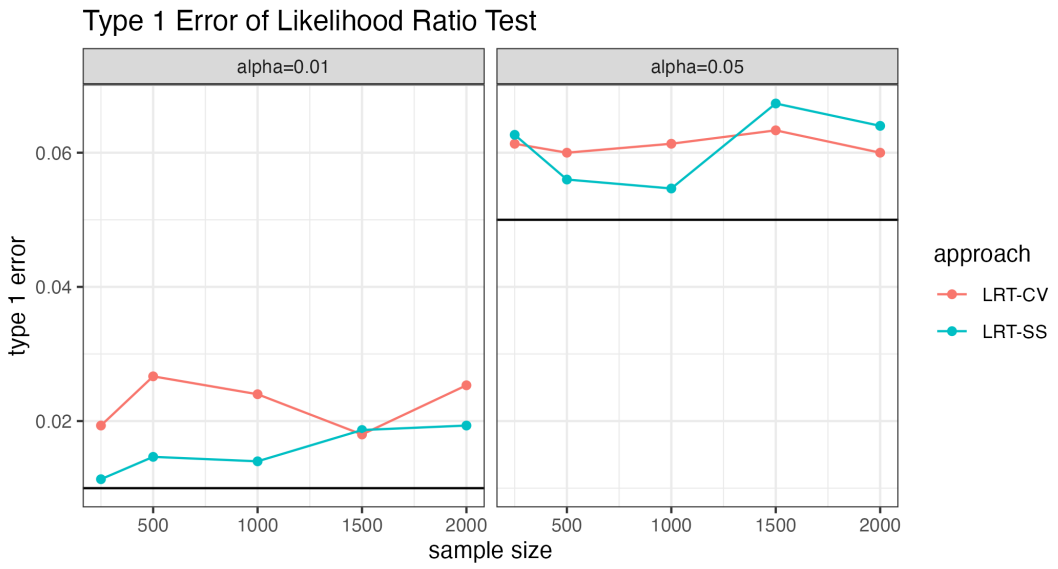


Figure 3.2: Type I error of likelihood ratio test applied to the targeted edge in various CPDAG structures. The black lines indicate the significance levels.

is documented in Figure 3.2. Overall, the type I error deviates minimally from the specified significance level and stabilizes as the sample size grows. Under both significance levels, the maximal difference between α and the type I error is within 0.015 for both the sample-splitting and cross-validation approaches. This also signifies that the test is robust against type I errors at smaller sample sizes, which can be attributed to the sample splitting design that renders independence between the estimated model and test data. The difference between the two approaches is minimal as well, differing by 0.005 at $n = 2000$, and indicates that both effectively control the false positive rate.

A type II error under the likelihood ratio test would be to falsely declare the two models as equivalent when only one model is true. In regards to edge orientation, that is to fail at identifying the true causal direction of an edge or incorrectly orient

an edge. For this experiment, we applied the likelihood ratio test to data generated from a singular non-linear function, where the orientation of the targeted edge is identifiable. We recorded the power of the test under significance level $\alpha = 0.05$.

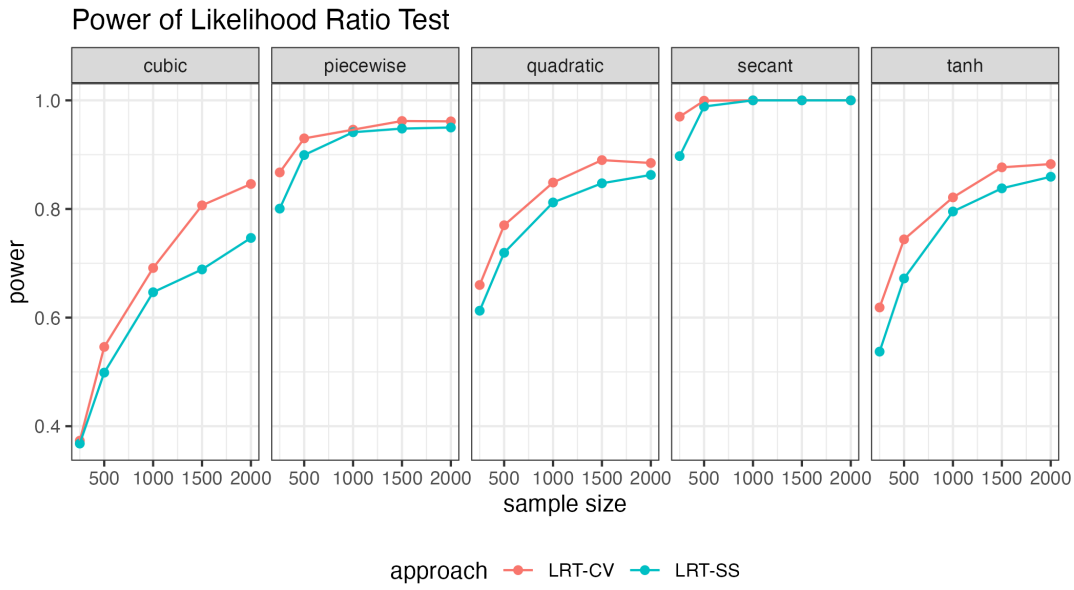


Figure 3.3: The statistical power of the likelihood ratio test on an undirected edge in a CPDAG, with select nonlinear functions underlying the SEM. Results show that power increases as n increases to at least 80%.

As seen in Figure 3.3, the statistical power of the likelihood ratio test increases with the sample size across all function types. For certain functions like the piecewise and secant functions, the power approaches one with $n \geq 1500$ samples. For other functions, the power improves more gradually but still strictly increases with the sample size, including the case of the cubic function. In particular, the power increases by at least 33% between sample sizes $n = 250$ and $n = 1000$. The cross-validation based approach exhibits greater power, achieving at least 85% by $n = 2000$ across all

function types. The results provide empirical evidence that the test can in practice identify the true causal direction of an undirected edge in a CPDAG, especially with sufficient data.

3.1.3 Comparison of Algorithm Performances

We evaluated the performance of SNOE against competing methods on synthetic data. We compared our method to CAM, NOTEARS, DAGMA, and SCORE, where each represents a different approach for learning nonlinear DAGs. CAM [Bühlmann et al., 2014] is a score-based method that assumes a nonlinear causal additive model (1.7) with Gaussian noise and optimizes the log-likelihood function to learn a DAG. Utilizing deep neural networks to model SEMs, NOTEARS [Zheng et al., 2020] and DAGMA [Bello et al., 2022] formulate the structural learning problem as a continuous-optimization problem with an algebraic constraint to enforce acyclicity. In the following experiments, we employed the version tailored to learning from nonlinear data for both algorithms. SCORE [Rolland et al., 2022] employs a bottoms-up approach to iteratively identify leaf nodes by computing the Jacobian of the score function under the assumption of a Gaussian error distribution. For all methods, we used their recommended parameter settings. While several constraint-based methods were tested as well, their performance fell short. A detailed analysis of their performances is provided in Appendix A.3.1.

The algorithms were applied to learn six DAG structures of varying sizes selected from the **bnlearn** network repository. For each network, we generated $N = 75$ data sets, each with a sample size of $n = 1000$, using the additive model with Gaussian noise in Equation 1.7. The SEMs were created under three separate functional forms:

linear functions, invertible functions, and non-invertible functions. The function classes are denoted respectively as *linear, inv,* and *ninv* in the figures. Under the invertible functions setting, we randomly selected functions from a set consisting of the cubic, inverse sine, piece-wise linear, and exponential functions. The non-invertible functions were sampled from a Gaussian process using a squared exponential kernel and bandwidth $h \sim \text{Unif}(5, 5.25)$. For all cases, the Gaussian noise term was sampled with mean $\mu = 0$ and standard deviation $\sigma \sim \text{Unif}(0.5, 0.75)$.

The true DAG serves as the ground truth for evaluation, with the exception of the linear, Gaussian case for which only the MEC is identifiable and thus the true CPDAG is used as the ground truth. Our main evaluation metrics are the F1 score, the structural Hamming distance (SHD), and the computational complexity. The F1 score is a harmonic mean of the precision and recall scores. It is calculated as $F1 = \frac{2TP}{2TP+FP+FN+2IO}$, where TP, FP, FN, and IO respectively denote the number of true positives, false positives, false negatives, and incorrectly oriented edges. The SHD measures the number of edge additions, deletions, and reversals required to convert the learned DAG into the true DAG. The computational complexity is measured by overall runtime in seconds.

The results presented in Figure 3.4 demonstrate that SNOE consistently outperforms competing methods. We observe that our algorithm achieved uniformly high F1 scores across all network structures and functional forms, with an average standard deviation of 0.06 in its performance across the three types of functions for both approaches. SNOE performed particularly well on invertible nonlinear DAGs, which presents a more challenging task due to the difficulty of detecting such nonlinear relations. On average across all function types, the F1 score of SNOE is respectively 67.6%, 61.8%, and 100.1% higher than those of NOTEARS,

DAGMA, and SCORE. A closer analysis reveals that NOTEARS and DAGMA produced sparser DAGs by missing considerable amounts of edges, while SCORE often predicted many extraneous edges without capturing the true edges. We also performed hyperparameter tuning on two learning rate parameters of SCORE for a wide range of values, $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$, and the resulting F1 scores were very similar for all combinations. CAM performed similarly to our method under data generated from non-invertible functions, but showed considerable variability based on the data-generating function. Its F1 scores for invertible and linear functions are lower than SNOE with sizable margins for most networks. Specifically, CAM produced denser DAGs with more false positives, especially under these settings.

Our method also ran significantly faster than the competing methods. We show the average runtime on the $\log_{10}$ scale against network size (number of nodes) in Figure 3.5. The sample-splitting approach (SNOE-SS) has a slightly shorter runtime than the cross-validation approach (SNOE-CV), since the latter performs regression twice. For the largest network, our algorithm was at least 7.7 times faster than all competing methods. While CAM showed similar F1 scores in certain cases, SNOE-SS was between 2.8 to 10.7 times faster than CAM, with the difference magnified when learning larger networks. This efficiency can be attributed to its local learning approach of identifying edges satisfying the PANM, rather than optimizing over the entire DAG. Further reduction in runtime occurs as Meek's rules typically orient additional undirected edges after performing the orientation test. In contrast to score-based and optimization-based methods that search over a restricted DAG space, our method leverages properties inherent to the graphical structure and only evaluates a sub-DAG, containing just the relevant nodes, to determine the correct edge orientations. Consequently, this results in higher computational efficiency for

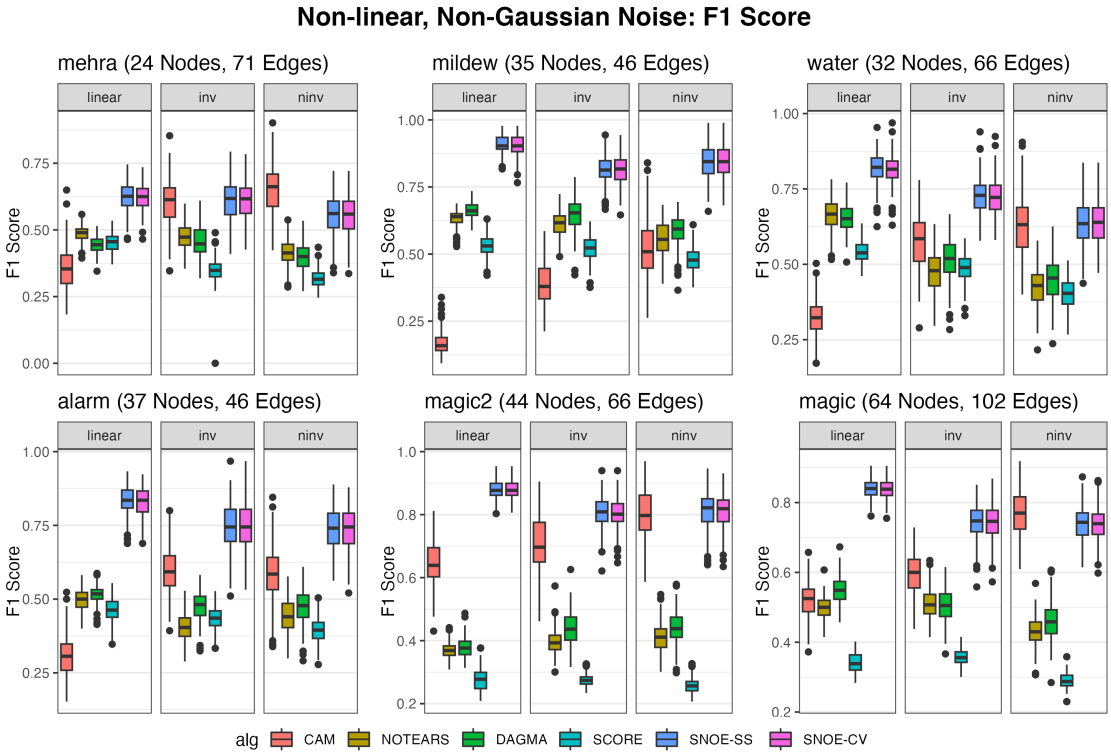


Figure 3.4: F1 score of learned graphs on simulated data generated under linear, invertible, and non-invertible functions with Gaussian errors.

structural learning.

We further investigated the empirical performance of our algorithm under model mis-specification, specifically when the noise distribution is non-Gaussian, and present results in Figure 3.6. Recall that the DAG is identifiable when the noise terms follow a non-Gaussian distribution. For this experiment, we simulated data under the same previous settings, but sampled the error variables from three different non-Gaussian distributions: the t-distribution with $df = 5$, the Laplace distribution, and the Gumbel distribution, all with $\mu = 0$ and $\sigma \sim \text{Unif}(0.5, 0.75)$. Since the learning accuracies for each error distribution are similar, we present the combined results.

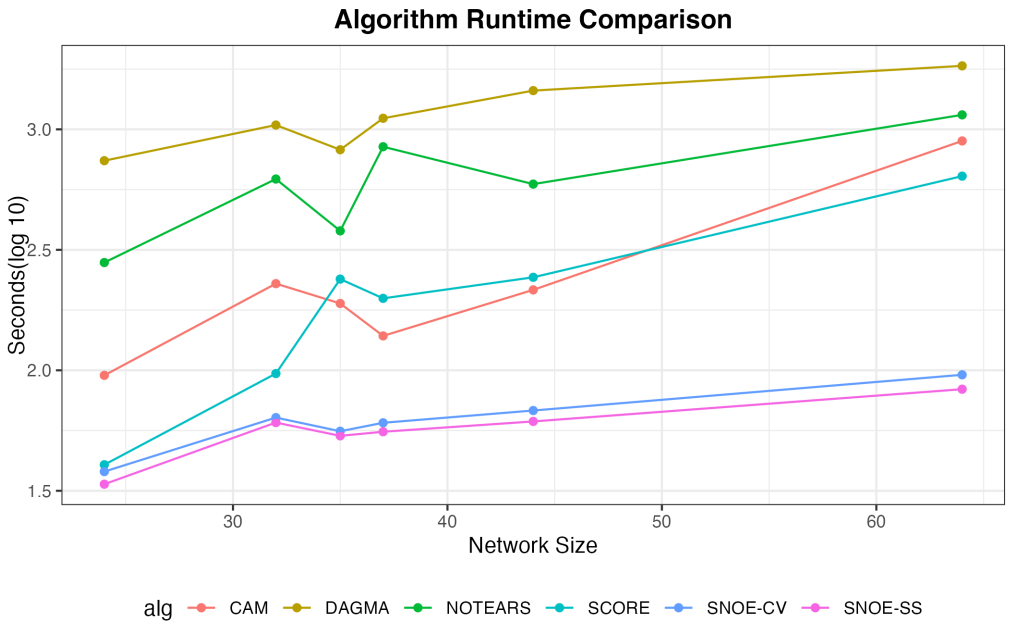


Figure 3.5: Average runtime in $\log_{10}$(seconds) of algorithms by network size.

Both variations of SNOE achieved higher accuracy than competing methods across all settings again, while only CAM was comparable in a few cases. Similar to the Gaussian case, the F1 score of our method is consistent under different function types with non-Gaussian noise. The exact F1 scores are comparable to the Gaussian case as well, indicating that our method is robust to model mis-specification and is a versatile causal learning method.

In addition, we examined the structure learning accuracy of our algorithm under data generating models that are not additive, such as multi-layer perceptrons or nonlinear SEMs including interactions terms. Results in Appendix A.3.2 suggest that SNOE is quite robust and still outperformed the competing algorithms.

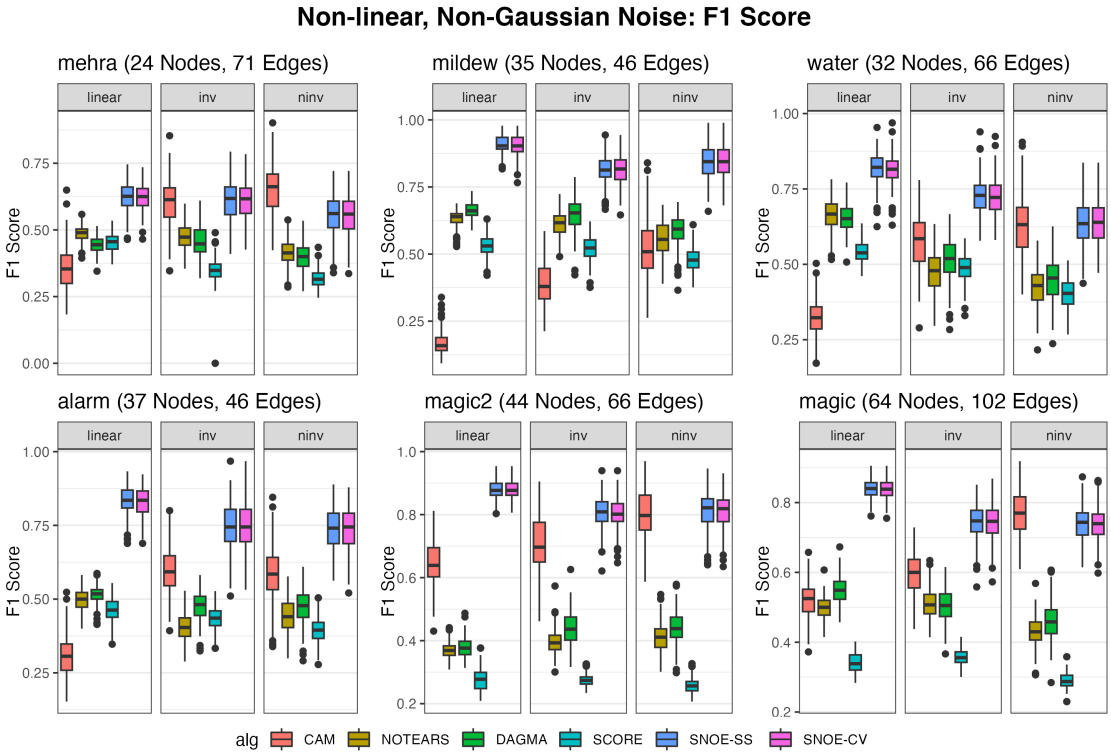


Figure 3.6: F1 score of learned graphs on simulated data generated under linear, invertible, and non-invertible functions with *non-Gaussian errors*.

3.1.4 Intermediate Results at Individual Stages

Having presented the performance of our algorithm against competing methods, we now closely analyze its accuracy after each of the following four stages in Algorithm 2: (1) *initial learning* to learn the initial PDAG, (2) *edge orientation* to determine the causal direction of undirected edges in the PDAG, (3) *edge deletion* to remove superfluous edges, and (4) *graph refinement* to extend the PDAG to a DAG, if applicable, by applying the edge orientation step again. As mentioned in Section 2.2, while our framework produces a PDAG in general, the implementation incorporates a fourth step to produce a DAG given the non-linear ANM assumption. Figure 3.7 shows the F1 score computed after each of these four stages. The algorithm's capabilities are best demonstrated in learning nonlinear DAGs, where the F1 score improves by 7.7% to 53.3% from the initial step to after the deletion step under nonlinear functions. See Appendix A.3.5 for analysis and discussion on the runtime of the intermediate steps.

It is imperative to recall that the initial graph is dense because it incorporates an extra candidate edge set (U_{α_2} on Line 10 in Algorithm 2), which may include some false positives. Although the deletion step appears to yield the greatest increase in the F1 score, the edge orientation step actually first uncovers more true positives from the undirected edges (see Figure A.6 in Appendix A.3.4). A detailed analysis shows that the number of true positives increases by 3.3% to 23.6% after applying the orientation procedure. The extra candidate edges may contain true positive edges or edges in the true DAG that are crucial to correctly orienting undirected edges; these edges would otherwise not be recovered in the latter stages. Since the number of true positives remains unchanged after the edge deletion step, we can conclude

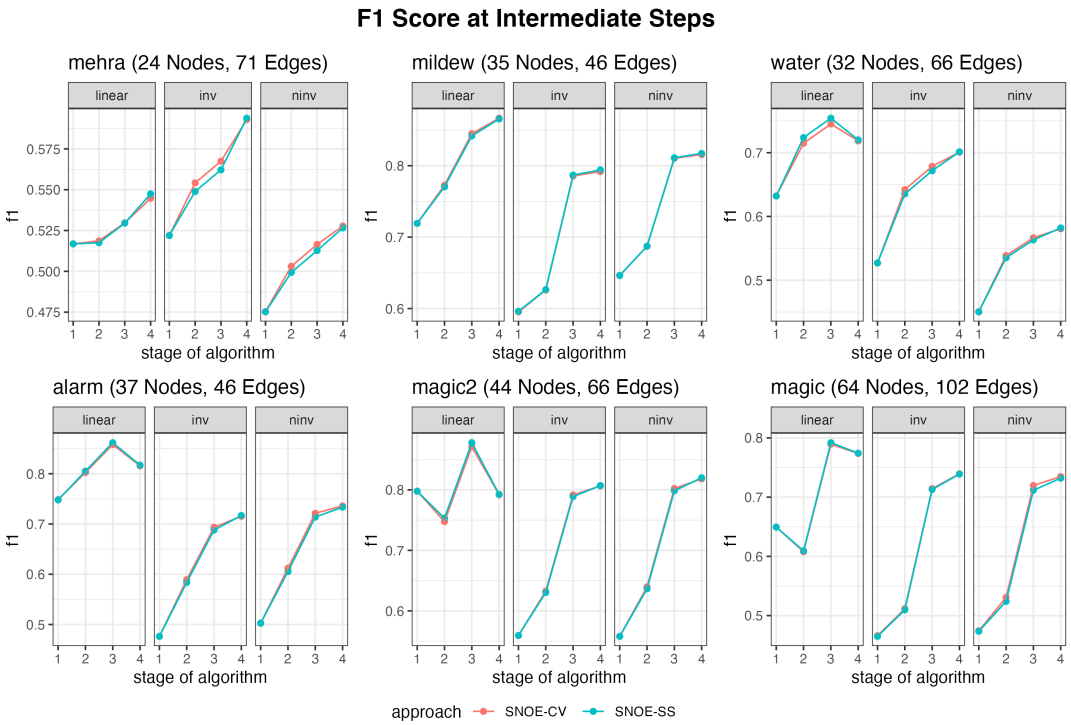


Figure 3.7: The F1 score after each stage of the framework: (1) initial CPDAG learning, (2) edge orientation, (3) edge deletion, and (4) graph refinement. The two curves overlap substantially in some panels.

that the deletion step correctly removes irrelevant edges. Therefore, the inclusion of additional candidate edges is essential and beneficial to our algorithm. Moreover, the significant increase of the F1 score from the initial stage evidences that SNOE can indeed enhance conventional algorithms that learn only an equivalence class.

Since in practice we do not know whether the nonlinear ANM is satisfied, we include the Gaussian linear case to show the performance when the identifiability result does not hold. The ground truth for evaluating this setting is the true CPDAG. In our algorithm design, we use the PC algorithm coupled with the partial correlation

test to learn the CPDAG, yielding the relatively high F1 score after the first stage in Figure 3.7. For several networks, the F1 score also increases after the second stage of orienting edges. In most practical applications, the estimated CPDAG is not perfect and thus may not capture all v-structures. The observed increase results from recovering missing compelled edges in the true CPDAG by the likelihood ratio test, as confirmed by the increase in true positives after stage two in these networks (see Figure A.6 in Appendix A.3.4). Since the first stage estimates a dense graph, the third stage removes false positives and thus improves the F1 score. The inclusion of candidate edges is helpful for the Gaussian linear case as well. Note that the final output is a DAG because we assume the nonlinear ANM; we then see a decrease in the F1 score after stage four which orients all remaining undirected edges. This is expected for Gaussian linear DAGs. Nevertheless, SNOE still exhibits high accuracy in this setting, demonstrating its versatility for causal learning under various functional forms.

Furthermore, we conducted an experiment to study the effect of misranking edges in the orientation procedure. We evaluated the undirected edges for orientation separately in ranked and arbitrary orders on the same initial graph, and then recorded the F1 score after the orientation stage. The results in Table A.2 in Appendix A.3.7 confirm the advantages of defining an order in our sequential method over a random search and the robustness against misranking. The analysis reveals that the ranking procedure indeed led to more true positives and higher F1 scores in the orientation stage. When edges are misranked, orienting an edge $X - Y$ that does not fulfill the PANM can lead to an erroneous orientation. However, the likelihood ratio test largely mitigates the impact of misranking since it may find models $X \to Y$ and $Y \to X$ indistinguishable or comparably fitted, thus leaving the edge undirected until

evaluation at a later iteration. See Appendix A.3.7 for a more detailed discussion.

We have also examined the impact of the initial graph on the performance of our algorithm (Appendix A.3.3). The results show that our edge orientation procedure can consistently recover the true DAG from estimated CPDAGs with varying levels of accuracy. In Appendix A.3.6, we provide insights on the impact of early errors during the orientation procedure; the experimental results suggest that the impact is minimal.

3.2 Real Data Applications

3.2.1 Flow-Cytometry Data

We applied all methods to the flow cytometry data set underlying the well-known Sachs network [Sachs et al., 2005]. The data set was collected in a study aiming to infer causal pathways amongst 11 phosphorylated proteins and phospholipids by measuring their expression levels after performing knockouts and spikings. Through a meta-analysis of both biological data and published literature, the researchers constructed a causal DAG that illustrates 17 causal relations among the 11 molecules. Given the high potential and broad applicability of causal learning in biological sciences, the Sachs network is one of the few verified causal graphs and is a popular means to benchmark causal learning methods. While the original data contains 7466 single-cell samples, we applied the algorithms to only the continuous version of the observational data set, which reduces the final data set to 2603 samples. It should be noted that the underlying skeleton is not fully connected; the graph consists of two disjoint clusters, one containing 8 nodes and the other containing 3.

Algorithm	Edges	SHD	F1	TP	FP	FN	Wrong Direction
CAM	19	19	0.39	7	9	7	3
NOTEARS	8	13	0.40	5	1	10	2
DAGMA	6	15	0.26	3	1	12	2
SCORE	17	20	0.29	5	8	8	4
SNOE-CV	**10**	**12**	**0.52**	**7**	**2**	**9**	**1**
SNOE-SS	11	13	0.50	7	3	9	1

Table 3.2: Algorithm performance on flow-cytometry data.

The performance summary of the estimated graphs in Table 3.2 shows the SNOE cross-validation approach produces a DAG closest to the ground truth, with a F1 score more than 30% higher than those of competing methods. Although its learned DAG is sparser than the true DAG, SNOE-CV has the highest F1 score and ties with SNOE-SS and CAM for the number of true positives captured. Despite missing several edges, it predicted very few false positives and thus has a lower SHD. The sample-splitting approach ranks second and differs from the cross-validation approach by predicting just one more false positive. NOTEARS and DAGMA were applied to both the original and standardized data, with the better results reported. Nevertheless, both methods still suffer from relatively high numbers of false negatives. Although CAM and SCORE have the densest DAGs predicted, they also have the highest counts of false positives and SHD.

3.2.2 Tübingen Cause-Effect Pairs

The Cause-Effect database is a collection of 108 different cause-effect pairs sourced from various domains such as biology, climate science, economics, and sociology

[Mooij et al., 2016]. Each data set consists of two variables with a known causal relation. The database has emerged as a popular benchmark for evaluating bivariate causal discovery methods given its diversity in data sources and data types. In this experiment, we applied both variants of the likelihood ratio test, the sample-splitting (SS) and cross-validation (CV) approaches, to each pair using a significance level of $\alpha = 0.05$. We consider 98 data sets since nine of them contain multi-dimensional variables.

Approach	# of undetermined cases	Accuracy when causal direction is determined	Overall accuracy
LRT-SS	63	68.6%	54.1%
LRT-CV	66	71.9%	66.3%

Table 3.3: Likelihood ratio test results on 98 cause-effect data sets.

The first two columns of Table 3.3 show the results of applying the likelihood ratio test with $\alpha = 0.05$ to determine the edge orientation. As shown in the first column, for more than 60 data sets, the causal direction cannot be determined by the likelihood ratio test at the significance level of 0.05. The causal models for $X \to Y$ and for $Y \to X$ were found to be equivalent for these data sets, suggesting that the SEMs may be approximated by linear models. Further analysis reveals that the average correlation between these undetermined pairs is 0.49, which signals a strong linear relation. For the other data sets, both approaches achieved a high accuracy in inferring the correct causal direction for about 70% of the data sets.

Mooij et al. [2016] applied their causal discovery methods to these data sets, assuming nonlinear ANMs as well. Their methods predicted the causal relations for all 98 data sets by choosing the direction with less dependence between the

residual and parent variables. To compare with their methods, we chose the direction having a larger log-likelihood value as the predicted causal direction for each data set, regardless of the test significance. We focus on their results for six entropy-based approaches, since the entropy measures are closely related to the normalized MI used in our work. Only one of their six approaches reached an accuracy of around 70% and the remaining all scored 40%–60%. As reported in the last column of Table 3.3, while LRT-SS exhibits a similar overall accuracy of 54.1%, LRT-CV outperforms most of their approaches with an overall accuracy of 66.3%. Mooij et al. [2016] attribute the large variation in accuracy to discretization effects when calculating the differential entropy. In our procedure, we normalize mutual information to avoid extreme values arising from distribution skewness or discretization. Furthermore, the MI measure is only used to identify edges for orientation by checking adherence to the PANM criterion, while the likelihood ratio test determines the causal relation in a robust way.

CHAPTER 4

Nonlinear Causal Discovery in the Presence of Latent Variables

4.1 Background

4.1.1 Acyclic Directed Mixed Graphs

Many scientific applications seek to infer causal relations from observational data, yet unmeasured variables are pervasive in practice, such as hidden biological pathways or socioeconomic factors, which may induce spurious associations that violate causal sufficiency. When causal sufficiency is violated, the conditional independence relations inferred from observed data no longer correspond to those implied in the true DAG, leading to several challenges [Colombo et al., 2012]. For example, in Figure 4.1(a), which depicts a DAG over all variables, the latent variable L_1 is a common cause of the observed variables X_2 and X_4. A constraint-based DAG learning algorithm applied only to the observed variables would generally infer a false adjacency between X_2 and X_4 because $X_2 \not\perp\!\!\!\perp X_4$, rather than the existence of a latent confounder. After marginalizing out latent variables, the resulting conditional independence structure among observed variables may no longer admit a DAG representation over the observed nodes alone. For instance, the conditional independence relation $X_2 \perp\!\!\!\perp X_4 \mid L_1$ cannot

be represented through d-separation after marginalizing over the latent variable.

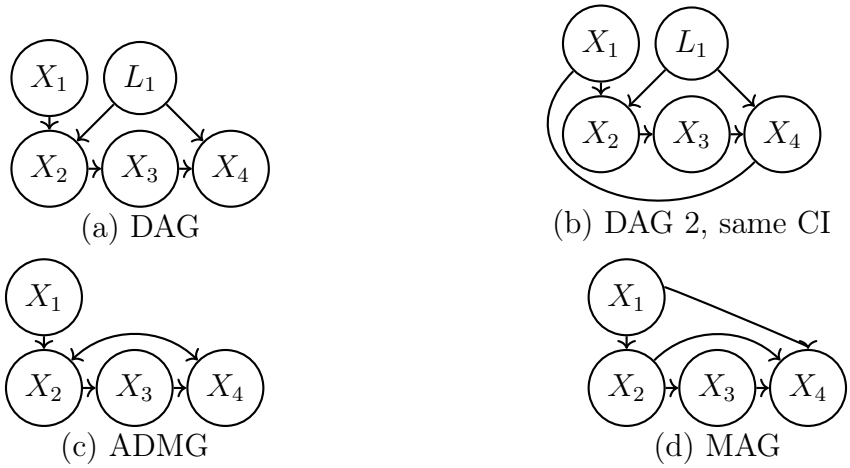


Figure 4.1: (a) A DAG with a latent variable L_1, (b) a graph with the same CI relations as the DAG that would be not distinguishable through CI tests only, (c) the ADMG showing relations over observed nodes, (d) the MAG encoding all conditional independence relations

To address this limitation, we instead use latent variable graphical models to represent relationships among observed variables. Let the vertex set of the true DAG $\mathcal{G}_0$ be partitioned as $V = (O, L)$, where O denotes observed nodes with variables $\{X_i\}_{i=1}^p = O$, and L denotes latent nodes. The latent projection of $\mathcal{G}_0$ onto the observed node set O can be represented by an *acyclic directed mixed graph* (ADMG) $\mathcal{G}$. An ADMG contains three types of edges: directed, undirected, and bidirected [Richardson and Spirtes, 2002]. A bidirected edge $i \leftrightarrow j$ indicates the presence of at least one latent confounder between i and j, making i a spouse (or sibling) of j, i.e. $i \in \text{sp}_{\mathcal{G}}(j)$ and vice versa. As shown in Figure 4.1(c), the ADMG includes the edge $X_2 \leftrightarrow X_4$ to represent the confounding effect induced by the unobserved variable L_1. An undirected edge $i - j$ indicates that i and j are both causes of an omitted

child node and independent in the true DAG. In this study, we assume the absence of selection bias, and therefore restrict attention to ADMGs without undirected edges.

A *directed path* is defined as a sequence of distinct vertices $V_1, \ldots, V_n$ such that each consecutive pair is connected by a directed edge pointing in the same direction. Node i is an ancestor of j (i.e., $i \in \text{an}(j)$), and j is a descendant of i (i.e., $j \in \text{desc}(i)$), if either $i = j$ or there exists a directed path from i to j. The concept of m-separation in latent variable graphical models generalizes d-separation in DAGs by additionally accounting for bidirected edges that arise from latent confounding. A path π between nodes i and j is said to be *m-connecting* relative to a set $S \subseteq V \setminus \{i, j\}$ if (1) every non-collider on π is not in S, and (2) every collider on π is an ancestor of at least one member of S. Nodes i and j are then *m-separated* by S if no path between them is m-connecting relative to S.

Bow-free ADMGs are a particular type of ADMG for which there is at most one edge between any pair of vertices. Beyond conditional independence relations, an ADMG may also impose equality or inequality constraints on the distribution $p(X)$ over observed variables $\{X_i\}_{i=1}^p$; these are known as Verma constraints Verma and Pearl [1990][Tian and Pearl, 2002]. For example, in the DAG shown in Figure 4.1(a), it can be demonstrated that

$$\sum_{X_2} P(X_4 \mid X_1, X_2, X_3) P(X_2 \mid X_1) = f(X_3, X_4) \tag{4.1}$$

is not a function X_1. As one can see, Verma constraints are implied by the underlying graph structure but are not generally expressible as conditional independence statements [Shpitser et al., 2014]. Moreover, the two DAGs in Figure 4.1(a) and (b) encode the same conditional independence relations, namely $X_1 \perp\!\!\!\perp X_4 \mid X_2$, as discussed in Tian and Pearl [2002]. However, without evaluating the equality statement in (4.1),

constraint-based methods cannot distinguish between these two graphs by conditional independence tests only.

4.1.2 Ancestral Graphs

In contrast, *maximal ancestral graphs* (MAGs) are a special class of ADMGs that provide a complete graphical representation of all conditional independence relations in the original DAG $\mathcal{G}_0$ through m-separation, while additionally preserving ancestral relations among observed variables [Zhang and Spirtes, 2012]. MAGs are termed *maximal* because for every pair of non-adjacent vertices, there exists at least one set of vertices that m-separates the pair. Assuming faithfulness to the DAG, for vertices i and j in MAG $\mathcal{G}$,

$$\text{i and j are m-separated by S in } \mathcal{G} \iff X_i \perp\!\!\!\perp X_j \mid S$$

for some conditioning set $S \subseteq V \setminus \{i, j\}$ in the distribution $p(X)$ over observed nodes. Thus, every missing edge corresponds to a conditional independence relation. MAGs are also *ancestral*, as they do not contain directed cycles nor almost directed cycles. An almost directed cycle occurs when $i \in \text{an}(j)$ while simultaneously $i \leftrightarrow j$.

As shown by Richardson and Spirtes [2002], every DAG $\mathcal{G}_0$ with latent variables can be uniquely transformed into a MAG over the observed variables. This transformation follows a marginalization procedure that preserves the conditional independence structure among observed nodes. First, two observed nodes i and j are made adjacent in the MAG $\mathcal{G}$ if there exists an inducing path between them relative to the latent variables in $\mathcal{G}_0$. Next, for each adjacent pair i, j, we orient $i \to j$ if $i \in \text{an}_{\mathcal{G}_0}(j)$ in the DAG, and orient $i \leftrightarrow j$ if neither node is an ancestor of the other. Figure 4.1(d) shows the MAG corresponding to the DAG in (a). In the MAG, the edge $X_1 \to X_4$

is introduced due to the presence of an inducing path between X_1 and X_4. Different from the ADMG representation, the edge between X_2 and X_4 is oriented as $X_2 \to X_4$ because $X_2 \in \text{an}_{\mathcal{G}_0}(X_4)$.

Just as multiple DAGs can encode the same set of conditional independence relations through d-separation, different MAGs may also imply the same set of conditional independence relations under m-separation, rendering them Markov equivalent. The corresponding Markov equivalence class can be represented by a *partial ancestral graph* (PAG), analogous to a CPDAG for DAGs, which captures the invariant edge marks shared across all MAGs in the equivalence class [Richardson and Spirtes, 2002]. A PAG may contain four types of edges: $\to$, $\leftrightarrow$, $\circ\!\!-\!\!\circ$, and $\circ\!\!\to$, defined through three possible endpoint marks: an arrowhead ($>$), a tail ($-$), and a circle ($\circ$) [Zhang, 2008a]. Arrowhead and tail marks in a PAG represent invariant edge marks shared by all MAGs in the equivalence class, whereas circle marks denote either an arrowhead or tail mark across equivalent graphs. Thus, PAGs provide a compact representation of both identifiable and uncertain structure under latent confounding.

In principle, a PAG can be fully recovered through performing conditional independence tests among observed variables [Zhang, 2008a], as distinct PAGs correspond to distinct sets of conditional independence constraints and therefore represent different Markov equivalence classes of MAGs. Under the causal Markov condition and faithfulness assumption, the Fast Causal Inference (FCI) algorithm, originally proposed by Spirtes et al. [2000], is widely regarded as the gold standard for PAG learning because it can consistently recover the true PAG given the oracle of conditional independence relations. The FCI algorithm is identical to the PC algorithm in the skeleton discovery and v-structure identification stages, except that all edges are initially represented with circle marks on both ends. These circle marks are changed through orientation.

During the orientation phase, FCI applies a distinct set of four orientation rules. The first three rules generalize Meek's orientation rules from DAGs to the latent variable setting, while the fourth rule exploits bidirected edges and collider structures along discriminating paths to orient an edge. While Ali et al. [2009] proved that these four rules are complete for identifying invariant arrowhead marks, Zhang [2008b] later extended this framework to ten orientation rules, establishing the completeness for identifying both invariant arrowhead and tail marks in a PAG.

4.2 Overview and Contributions

4.2.1 Overview

In Chapter 2, we focused on learning a DAG from its Markov equivalence class in the observational setting by developing the PANM criterion to identify edges that can be correctly oriented. In many real-world applications, however, causal sufficiency is often violated because not all relevant variables are observed. Unobserved variables can induce latent confounding between observed variables, creating dependence structures that cannot be adequately represented by a DAG over the observed variables alone. This limitation necessitates a richer graphical framework capable of capturing more complex relationships in the presence of hidden variables. We now turn our attention to learning graphical representations among observed variables under latent confounding.

There are several frameworks for modeling relationships among observed and latent variables, as well as those for learning such a graphical representation [Richardson and Spirtes, 2002][Shpitser et al., 2014][Evans, 2016]. As described in Chapter 4.1, an ADMG represents the latent projection of a DAG, where a bidirected edge $i \leftrightarrow j$

denotes the presence of one or more latent common causes of i and j. Among ADMGs, maximal ancestral graphs (MAGs) form a particularly important subclass because they encode all conditional independence relations among observed variables. This maximal property makes MAGs more informative and often more interpretable graphical structures. Partial ancestral graphs (PAGs), in turn, represent the Markov equivalence class of MAGs, using circle marks ($\circ$) to denote edge endpoints whose orientation varies across Markov equivalent MAGs. ADMGs are typically learned through score-based methods because they may also need to satisfy Verma constraints [Tian and Pearl, 2002], which are generally not identifiable through CI tests alone. In contrast, PAGs are most commonly learned through constraint-based procedures, with the FCI algorithm serving as the canonical approach.

In this work, we develop a framework for learning the ADMG of a true DAG under a nonlinear SEM, using the PAG as the starting graph. The framework is designed to first infer edge directions through conditional independence testing on subsets of edges that can be reliably oriented from graphical structure, and then determine the remaining ambiguous orientations through a procedure that iteratively estimates and compares log-likelihoods under possible candidate directions. By combining constraint-based identification with likelihood-based model comparison, this approach leverages the strengths of PAG-based causal discovery while introducing a principled mechanism for further determining orientations beyond standard equivalence-class information.

4.2.2 Related Work

Early work by Brito and Pearl [2002] established the identifiability of parameters under a linear Gaussian SEM given an ADMG $\mathcal{G}$, laying the foundation for subsequent structure-learning methods in latent variable settings. To evaluate the fit of a bow-free ADMG to observational data, Drton et al. [2009] developed an algorithm that estimates model parameters by iteratively fitting each variable conditional on its parent and spouse variables under the assumed linear Gaussian SEM, with the objective to maximize the log-likelihood. Nowzohour et al. [2017] proposed a greedy search procedure that constructs an ADMG by selecting graph structures with the highest penalized log-likelihood. By exploiting the district factorization property of ADMGs [Richardson, 2009], their method computes scores locally at the district level rather than over the full graph, substantially improving computational efficiency.

Alongside score-based ADMG learning, substantial efforts have advanced MAG and PAG discovery beyond the original FCI algorithm. These advances primarily target more accurate skeleton estimation and additional edge orientation through exploiting graphical patterns. For example, Ogarrio et al. [2016] proposed a hybrid strategy that first learns a CPDAG using DAG discovery methods, then applies the FCI algorithm to remove spurious edges introduced by latent confounding and to revise orientations based on newly identified m-separation relations. Separately, Pellet and Elisseeff [2008] demonstrated that constructing the graph skeleton through local Markov blanket discovery, rather than pruning a complete graph using repeated conditional independence testing as in FCI, can substantially improve both efficiency and orientation accuracy. This local-first perspective has since been extended in more recent PAG learning methods, including Li et al. [2025] and Ling et al. [2025], which

further leverage localized structural information to recover additional edge marks.

Building upon the identifiability results for nonlinear SEMs, recent work has extended latent causal discovery beyond linear settings to recover the latent projection of a DAG under more flexible functional assumptions. In Maeda and Shimizu [2021], the authors characterize identifiability of causal relations among observed variables under the CAM framework (1.7). They show that the causal direction between two adjacent observed nodes becomes unidentifiable when latent variables are present in frontdoor or backdoor paths between them. Based on this insight, they derive testable conditional independence relations on regression residuals, which form the basis of their CAM-UV algorithm for causal discovery with unobserved variables. Developing on these ideas, Ashman et al. [2023] establish identifiability results for ADMGs under a nonlinear SEM framework, providing sufficient assumptions under which the latent projection can be recovered from observed data. They also propose a deep learning-based, variational framework for estimating the ADMG $\mathcal{G}$ by maximizing an evidence lower bound (ELBO) for the observed-data likelihood $p(X \mid \mathcal{G})$.

4.2.3 Contributions

Although prior work has introduced important learning methods, existing approaches often remain constrained by restrictive model or graphical assumptions. Many classical methods rely on linear Gaussian SEMs, which can be inadequate for capturing the nonlinear relationships commonly observed in real-world systems. Under such misspecification, parameter estimates may fail to accurately infer directed and bidirected edge directions. Among the few nonlinear frameworks for ADMG, MAG, and PAG learning, many still depend heavily on repeated conditional independence

testing. Conditional independence tests can suffer from low power in finite samples, particularly in high-dimensional settings, and their performance may depend on the order of execution [Shah and Peters, 2020]. In addition, several existing nonlinear latent-variable methods require strong assumptions about the hidden structure itself, such as latent variables exerting dominant effects on observed variables or satisfying specific constraints on the number of children per latent node.

To this end, we propose a framework for learning the ADMG under the nonlinear SEM in this work. Given the PAG, our method aims to infer the edge marks of all undetermined marks by evaluating and orienting edges sequentially. Our framework differentiates between edges that can be correctly oriented and those that cannot. To orient such edges, we employ a constraint-based strategy that first tests for potential latent confounding and then applies the likelihood ratio test developed in Section 2.3.3 to distinguish the edge orientation. For the other edges, we compare the graph structure under different edge directions using estimated log-likelihood values obtained through an iterative parameter fitting procedure. The key to this procedure is to accurately capture the covariance term, which indicates the presence of a bidirected edge. Through this orientation framework, our method is able to infer substantially more edge marks in the PAG, thereby moving beyond standard equivalence-class information. Extensive numerical experiments demonstrate that our framework produces more accurate graph estimates than competing nonlinear ADMG and DAG learning methods across a wide range of functional forms, network sizes, and extent of latent confounding. Moreover, our approach consistently yields more informative graphs than both the exact PAG and estimated PAG alone, showing a substantial ability to refine partially identified structures and infer additional causal relations.

4.3 Preliminaries

4.3.1 Model Definition and Assumptions

Let $\mathcal{G}_0 = (V, E)$ be a DAG, where the vertex set $V = X \cup L$ consists of observed variables $\{X_i\}_{i=1}^p$ and unobserved variables $\{L_j\}_{j=1}^m$. We assume the data generating process follows a causal additive model (CAM) (1.7) given by

$$V_k = \sum_{i \in \mathrm{pa}(k) \cap X} f_{ik}(X_i) + \sum_{j \in \mathrm{pa}(k) \cap L} g_{jk}(L_j) + \varepsilon_k, \quad k = 1, \ldots, p+m \tag{4.2}$$

where both f_{ij} and g_{jk} are three times differentiable nonlinear functions and the error terms $\varepsilon_k \sim N(0, \sigma_k^2)$ are independent with $0 < \sigma_k^2 < \infty$. As demonstrated in Peters et al. [2014], the true DAG $\mathcal{G}_0$ can be identified from the joint distribution $p(V_1, \ldots, V_{p+m})$ under causal sufficiency and faithfulness.

As discussed, ADMGs represent causal relations among observed nodes using directed edges $\to$ and latent confounding using bidirected edges $\leftrightarrow$. Ashman et al. [2023] note that the corresponding "magnified" structural causal model—where each bidirected edge $i \leftrightarrow j$ is represented by introducing a latent variable h such that $i \leftarrow h \to j$ —is generally not unique, since latent variables may be shared across multiple observed variables. For instance, in the structure $i \leftrightarrow j \leftrightarrow k$, the nodes may share one latent node h confounding all nodes or multiple distinct latent variables confounding each pair, such as $i \leftarrow h_1 \to j$ and $j \leftarrow h_2 \to k$. In this work, our objective is therefore to recover the latent projection over observed variables rather than the exact latent structure in the DAG over $V = (X \cup L)$.

We further assume that all latent variables are root nodes in $\mathcal{G}_0$, therefore each $L_j \in L$ satisfies $L_j = \varepsilon_j$. This also implies joint independence among the latent variables. Moreover, each latent variable is assumed to have exactly two unique,

non-adjacent observed children, corresponding to a configuration $i \leftarrow h \rightarrow j$. Under these assumptions, the latent projection over observed variables is a bow-free ADMG, where each pair of observed nodes shares only one edge type.

4.3.2 Identifiability of ADMG

As discussed in Chapter 4.2.2, prior works have characterized both the identifiability of causal relations under the CAM in (4.2) with latent variables and the identifiability of the ADMG under certain conditions. Maeda and Shimizu [2021] generalize the latent confounding structure as an unobserved backdoor path (UBP), defined as a path starting with i and its unobserved cause m and ending with j and its unobserved cause n, i.e. $i \leftarrow m \leftarrow \ldots \leftarrow k \rightarrow \ldots \rightarrow n \rightarrow j$, where k is a common ancestor. Under the assumption of the cascade ANM [Cai et al., 2019], the causal effect of i on j on path $i \rightarrow h \rightarrow j$, where h is a hidden variable, cannot be determined by residual independence, i.e. $X_j - f(X_i) \not\perp\!\!\!\perp X_i$. We refer to their findings to characterize the presence of a latent confounder in a mixed graph:

Lemma 2. *(Lemma 1 of Maeda and Shimizu [2021]) Assume the data generation process is* (4.2)*. If and only if equation* (4.3) *is satisfied, there is a UBP between X_i and X_j where f_i and f_j denote regression functions.*

$$\forall f_i, f_j, M \subseteq (X \setminus X_i), N \subseteq (X \setminus X_j), \quad [X_i - f_i(M)] \not\perp\!\!\!\perp [X_j - f_j(N)]. \tag{4.3}$$

Lemma 2 implies that if an unobserved backdoor path exists between i and j, then the residual of X_i regressed on any subset of $X \setminus X_i$ and the residual of X_j regressed on any subset of $X \setminus X_j$ cannot be mutually independent. In particular, it is only assumed that $f_\cdot$ is nonlinear in the CAM (4.2). This statement naturally

provides a testable criterion for determining the existence of an unobserved common cause between two variables. Conversely, stated in Lemma 2 of Maeda and Shimizu [2021], if the residuals of X_i and X_j, constructed separately from any subsets of $X \setminus \{X_i, X_j\}$, are independent, then no unobserved backdoor path exists between them. Thus, the conditional independence between residuals implies that the pair is non-adjacent in the graph over observed nodes.

Expanding upon this insight, Ashman et al. [2023] establish the identifiability of ADMGs from observational data under nonlinear SEMs subject to two key assumptions. First, the data generating function is assumed to follow the model

$$V_k = f_{i \in pa(k) \cap X}(X_i) + g_{j \in pa(k) \cap L}(L_j) + \varepsilon_k$$

where the functions f, g are nonlinear and error terms $\{\varepsilon_k\}_{i=1}^{p+m}$ are jointly independent. The covariates may follow a general nonlinear SEM as in (1.5), while requiring the effects from observed and latent variables to be separable. Second, each latent variable is assumed to be exogenous and to confound exactly one unique pair of non-adjacent observed nodes. Every bidirected edge in the ADMG then corresponds to a distinct latent common cause, thus eliminating the possibility of a shared latent structure. The researchers show that given these assumptions, the conditional independence relations in Maeda and Shimizu [2021] are also sufficient to determine the edge orientation in the ADMG.

4.4 Methodology

For causal discovery without causal sufficiency, classical constraint-based methods are generally limited to recovering a PAG, as they rely solely on conditional independence

relaions and do not evaluate Verma constraints [Verma and Pearl, 1990], which are necessary for identifying the ADMG. Although score-based approaches can, in principle, be applied to learn MAGs or the broader class of ADMGs, they typically require performing a computationally intensive procedure of repeatedly calculating graph scores across the search space and are not guaranteed to attain the global optimum. Our objective is instead to learn a graphical representation over the observed nodes by determining further edge marks in the PAG. In particular, our framework distinguishes between edges whose orientations are identifiable by a single iteration of conditional independence tests and those that are not. It then applies a separate orientation strategy to each edge type, allowing for more efficient search through a constraint-based method and more comprehensive comparisons through a score-based method. In this section, we introduce the criterion used to differentiate these two categories of edges and present the complete sequential orientation procedure.

4.4.1 Graphical Criterion for Orientation

The defining component in the DAG learning algorithm by Huang and Zhou [2025] is the PANM criterion (2), which is used to identify undirected edges that can be correctly oriented at a given stage of the algorithm. In the presence of latent variables, however, there is no guarantee that such an edge can be correctly oriented at every stage. Consequently, we cannot create a complete evaluation order of undetermined edges, and the correctness of the likelihood ratio test depends on whether this criterion is satisfied. While a latent confounder can be identified through iterative testing, this procedure is computationally intensive and requires careful adjustment of the significance level.

To address these limitations, we develop an alternative sound graphical characterization to identify whether the orientation of (i, j) can be correctly inferred using conditional independence tests in the given graph. In a mixed graph, i and j may share an unidentified observed common parent. Our graphical characterization excludes the possibility that the confounding relation between i and j arises from unidentified common causes among observed variables. In other words, it distinguishes edges whose non-identifiability is attributable only to latent confounding, rather than undetermined relations with their neighboring nodes that would be missing in the SEM constructed according to the graph. A key advantage of this criterion is that it depends solely on the graph structure and is therefore independent of the data and sample size, allowing such edges to be identified solely from the graphical structure.

Suppose $\mathcal{G}_0$ is the true DAG and $\mathcal{G}$ is a partial mixed graph, where $\mathcal{G}$ can be converted into the latent projection of $\mathcal{G}_0$ through a series of edge orientations. We introduce the following graphical characterization to identify edges in $\mathcal{G}$ for which residual dependence can only be explained by latent confounders:

Proposition 2. *(Graphical Characterization for Conditional Independence Tests) Suppose i and j are two adjacent nodes in a partial mixed graph $\mathcal{G}$ where (i, j) contains at least one circle mark. Then the true edge orientation of (i, j) can be evaluated using conditional independence tests if one of the following holds:*

(i) For all edges connecting to nodes i or j, excluding (i, j), (1) every arrowhead into i or j has no circle endpoint (ex: $k \to, \leftrightarrow i$) and (2) every circle endpoint at i or j has an arrowhead endpoint (ex: $i \circ\!\!\to k$).

(ii) Node i m-separates j from $ne_{\mathcal{G}}(i) \setminus \{j\}$ and $pa_{\mathcal{G}}(i) = \emptyset$, while j satisfies condition (i).

Condition (i) corresponds to the case where the true parent sets $\text{pa}_{\mathcal{G}}(i)$ and $\text{pa}_{\mathcal{G}}(j)$ are fully identified in the mixed graph $\mathcal{G}$. Equivalently, neither i nor j has any remaining unidentified parent nodes in $\mathcal{G}$. If the true causal relation in $\mathcal{G}$ is $i \to j$, then condition (i) is satisfied whenever the parent sets $\text{pa}_{\mathcal{G}_0}(i)$ and $\text{pa}_{\mathcal{G}_0}(j) \setminus \{i\}$ are recovered in $\mathcal{G}$. By contrast, if the true orientation is $i \leftrightarrow j$, then any confounding relationship between i and j must arise from an unobserved common cause, since all observed parents, including shared parents, in $\mathcal{G}$ have already been identified.

Figure 4.2(a) illustrates this case: edge (i, j) satisfies condition (i) because both nodes have no unidentified parents. Nodes i and j share a common parent A, while j additionally has parent B, and the edge $i \circ\!\!\to C$ implies that C can only be a child or sibling of i. If the true causal direction is $i \to j$, then the causal direction can be determined because the SEMs $X_i = f(A) + \varepsilon_i$ and $X_j = g(A, B) + h(X_i) + \varepsilon_j$ are correctly recovered from $\mathcal{G}$, and thus the error terms satisfy the relation $\varepsilon_i \perp\!\!\!\perp \varepsilon_j$. The same reasoning applies for $j \to i$. Otherwise, if $\varepsilon_i \not\perp\!\!\!\perp \varepsilon_j$, this dependence must be attributed to a latent confounder $i \leftarrow L \to j$. Edge (i, j) in (b) does not satisfy condition (i) because B may be a parent of j and C may be a parent of i. The SEMs for X_i and X_j implied by $\mathcal{G}$ would now omit the parent variables, resulting in $\hat{\varepsilon}_i \not\perp\!\!\!\perp \hat{\varepsilon}_j$. In this case, it becomes impossible to determine whether the residual dependence arises from unidentified parents in $\mathcal{G}$ or from a latent confounder.

Condition (ii) applies to an edge (i, j) when one node, say j, satisfies condition (i), while the other node i may still have possible parent nodes among its neighbors in $\mathcal{G}$. Since i has no identified parents, we can write the SEM for X_i as $X_i = \tilde{\varepsilon}_i = \sum_{k \in \text{pa}_{\mathcal{G}}} f_{ik}(X_k) + \varepsilon_k$, where the error term $\tilde{\varepsilon}_i$ absorbs both ε_i and all unidentified parents of i. The m-separation condition between j and $\text{ne}(i)$ by i prevents unidentified parent nodes of i from inducing residual dependence with j through active collider

paths. Although the full parent set of i may not yet be identified, the edge (i, j) can still be evaluated. Under the true causal direction $i \to j$, $pa_{\mathcal{G}}(j) \setminus \{i\}$ is fully identified and we can expressed all causes and the error of X_i into a single term to find $\varepsilon_i \perp\!\!\!\perp \emptyset$ and $\varepsilon_j \perp\!\!\!\perp \{\mathrm{pa}_{\mathcal{G}}(j) \cup i\}$. If the true relation is $i \leftrightarrow j$ instead, then any residual dependence must be induced by a latent confounder, because the undetermined observed parents of i in $\mathcal{G}$ have already been absorbed into $\tilde{\varepsilon}_i$. In this sense, condition (ii) is a relaxed version of condition (i): one node may still have unidentified parents, provided these unidentified causes do not create additional uncertainty.

Figure 4.2(c) illustrates this case. Node j has all parent nodes recovered except possibly i, while node i m-separates j from neighbors A and B of i. Therefore, if the true direction is $i \to j$ or $i \leftrightarrow j$, the edge orientation can be determined through SEMs $X_j = f(C, D) + \varepsilon_j$ and $X_i = \tilde{\varepsilon}_i$, where the error term ε_i and possible parent variables are merged into $\tilde{\varepsilon}_i$. By contrast, Figure 4.2(d) does not meet this condition. When the local structure becomes $j \circ\!\!\to i$, node i acts as a collider and causes paths $j \circ\!\!\to i \leftarrow\!\!\circ A$ and $j \circ\!\!\to i \leftarrow\!\!\circ B$ to become m-connecting. By the same reasoning as in Figure (b), dependence between residuals of i and j can no longer be attributed uniquely to a latent confounder, so the edge orientation cannot be correctly determined.

In essence, this graphical characterization allows us to identify edges in $\mathcal{G}$ for which residual dependence or independence can be meaningfully interpreted. Namely, it allows us to orient bidirected edges induced confounded by a latent node rather than by unidentified parent nodes in $\mathcal{G}$. Although the proposed graphical characterization may not be exhaustive, it is structurally sound given a correct mixed graph. It provides a deterministic and data-independent mechanism for prioritizing edges whose orientation can be evaluated with substantially greater reliability, in the case of limited samples and multiple testing.

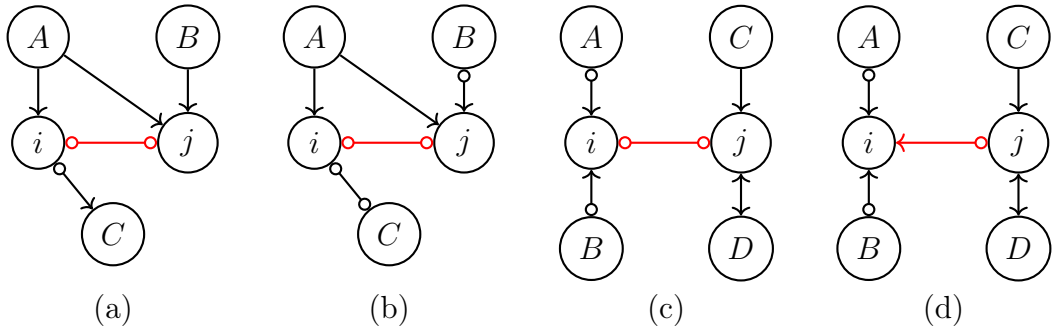


Figure 4.2: Examples to illustrate the graphical criterion in Prop. 2 for edge (i, j). (a) the edge satisfies condition (i) because $\text{pa}_{\mathcal{G}}(i)$ and $\text{pa}_{\mathcal{G}}(j)$ are fully identified; (b) the edge does not satisfy the criterion since both nodes have possible parent nodes; (c) the edge satisfies condition (ii), as i m-separate j from its neighbors, while $\text{pa}_{\mathcal{G}}(i) = \emptyset$ and $\text{pa}_{\mathcal{G}}(j)$ is fully identified; (d) the edge does not satisfy the criterion because $j - i - A$ and $j - i - B$ are now m-connecting paths, making j, A, B all possible parents of i.

4.4.2 Algorithm Overview

The complete algorithm is formally presented in Algorithm 5, detailing a sequential edge orientation procedure for causal discovery under latent variables. The ADMG learning task achieved through two stages: (1) estimating an initial PAG using the FCI algorithm, and (2) sequentially orienting edges that contain circle marks. In practice, we simplify the first stage by applying only a subset of the full PAG orientation rules, producing a partial mixed graph whose arrowhead and tail edge marks may be incomplete but remain sound. Practitioners may alternatively choose to apply the full set of orientation rules if a more complete PAG is desired, as well as a deletion step if desired.

We begin with applying a modified version of the FCI algorithm [Spirtes et al.,

2000], summarized on Lines 1–10. Starting with the complete graph, the algorithm first estimates the skeleton by performing conditional independence tests and removing an edge (i, j) whenever $i \perp\!\!\!\perp j \mid S$ for some conditioning set $S \subseteq V \setminus \{i, j\}$ at significance level α_1 (Lines 5–7). Specifically, we use the kernel-based Hilbert–Schmidt Independence Criterion (HSIC) test proposed by Gretton et al. [2007] to better evaluate nonlinear relations. After estimating the skeleton under α_1, we identify v-structures and apply FCI orientation rules 1–3 to produce a partially mixed graph (Lines 8 and 9). To avoid missing true edges in the graph, we first construct a denser candidate graph using a relaxed significance threshold α_2 (Lines 1–3), where typically $\alpha_2 > \alpha_1$. The stricter threshold α_1 is then applied in the subsequent stage to refine the skeleton used for orientation, yielding more precise v-structures and inference on edge marks. Once orientations are determined under α_1, edges retained under α_2 but removed under α_1 are reintroduced as fully undetermined edges, denoted as $\circ\!\!-\!\!\circ$, on Line 10. The result is a partial mixed graph with high precision of edgemarks containing arrowheads and tails (using the graph estimated by α_1) while retaining additional plausible edges for later evaluation.

To determine edges containing circle marks, we differentiate between two types of undetermined edges. If an edge (i, j) fulfills the graphical criterion in Proposition 2, then its SEM over observed variables can be correctly estimated. The intuition behind this criterion is that it accounts for all possible parent nodes of a pair of nodes, so the dependence between residuals can be attributed to a latent confounder rather than to undetermined parent nodes among observed variables. We refer to this subset as U_{CI}, which is identified on Line 13. These edges are prioritized because their local graphical structure permits straightforward application and interpretation of conditional independence tests.

Algorithm 5: Finite-sample version of algorithm

Input: Complete graph $\mathcal{G} = (V, E)$, observed data $X = \{X_i\}_{i=1}^{p}$, sig. level α_1, α_2 where $\alpha_1 < \alpha_2$

Output: ADMG $\mathcal{G} = (V, E)$

for $(i, j) \in E$ **do**
  Search for separating set $S_{ij} \subseteq V$ such that $p\text{-}val(X_i \perp\!\!\!\perp X_j | S_{ij}) > \alpha_2$;
  Update $E \leftarrow E \setminus \{(i, j), (j, i)\}$ and store S_{ij} if found;
$E_{\alpha_2} \leftarrow E$;
for $(i, j) \in E$ **do**
  Search for separating set $S_{ij} \subseteq V$ such that $p\text{-}val(X_i \perp\!\!\!\perp X_j | S_{ij}) > \alpha_1$;
  Update $E \leftarrow E \setminus \{(i, j), (j, i)\}$ and store S_{ij} if found;
Detect v-structures given E and $\{S_{ij}\}$;
Orient remaining edges by $R_1 - R_3$ of the FCI algorithm orientation rules;
Merge candidate edge set $U_{\alpha_2} \leftarrow E_{\alpha_2}$ to obtain $\mathcal{G} = (V, E \cup U_{\alpha_2})$;
Obtain $U \leftarrow \{U_i : U_i \in \{\circ\!\!\rightarrow, \circ\!\!-\!\!\circ\}\}$;
while $|U| > 0$ **do**
  Identify edges U_{CI} by Prop. 2;
  for $(i, j) \in U_{\text{CI}}$ **do**
    $\mathcal{G} \leftarrow \text{OrientByCI}\{\mathcal{G}, (i, j), \alpha, X\}$ (Algorithm 6);
  $U_{-\text{CI}} \leftarrow U \setminus U_{\text{CI}}$;
  Calculate the number of common neighbors $|\text{ne}(U_k)|$ in $\mathcal{G}$ for $U_k \in U_{-\text{CI}}$;
  for $i = 0, \ldots, \max\{|\text{ne}(U_k)|\}$ **do**
    for $(i, j) \in \widetilde{U}$, *where* $\widetilde{U} = \{U_k \in U : |ne(U_k)| = i\}$ **do**
      $\mathcal{G} \leftarrow \text{OrientByScore}\{\mathcal{G}, X, (i, j)\}$ (Algorithm 7);
  Update $U \leftarrow \{U_i : U_i \in \{\circ\!\!\rightarrow, \circ\!\!-\!\!\circ\}\}$;

We develop the procedure OrientByCI to determine the edge marks of a candidate edge, as described in Algorithm 6. We use the likelihood ratio test developed in Section 2.3.3 to determine the causal direction. However, the likelihood ratio test alone is insufficient for latent variable settings: failure to reject H_0 implies only that the competing models are observationally equivalent, instead that no latent confounder is present. Moreover, valid inference of the test requires no latent confounders exist. Therefore we first evaluate whether the pair is confounded by testing the fit of regression models for possible causal directions $i \to j$ and $j \to i$. Under $i \to j$ for instance, we estimate

$$X_i = \sum_{k \in \text{pa}(i)} f_{ik}(X_k) + f_{ij}(X_j) + \varepsilon_k,$$

where j is treated as a parent of i. The model for $j \to i$ is defined similarly. As shown on Line 5, we then assess whether the estimated residual from either model is dependent with any included covariate. As the edge satisfies our graphical criterion, residual dependence would imply that a latent confounder exists between i and j. The algorithm will then orient the edge as $i \leftrightarrow j$. If one candidate causal direction yields residual independence while the alternative does not, then the data is more consistent with that orientation under the assumed SEM. For a partially oriented edge $i \circ\!\!\to j$, only one causal direction is possible, so we test for residual independence in the SEMs under $i \to j$ and orient accordingly (Line 6). For a fully undetermined edge $i \circ\!\!-\!\!\circ j$, both directions remain possible, so we apply the likelihood ratio test to compare the two candidate directions directly (Line 9). To reduce overfitting and preserve valid inference, the algorithm performs a train-test split on the dataset, where regression model estimators are constructed from the training data and test statistics are estimated from the test data.

Algorithm 6: Constraint-Based Edge Orientation Procedure (OrientByCI)

Input: Graph $\mathcal{G} = (V, E)$, edge (i, j), sig. level α, observed data

$$X = \{X_i\}_{i=1}^{p}$$

Output: Edge (i, j)

1 Perform train-test split on observed data;

2 Fit models $\hat{X}_i = \sum_{k \in \mathrm{PA}_i} f_k(PA_i) + f_j(X_j)$ and $\hat{X}_j = \sum_{h \in \mathrm{PA}_j} g_h(PA_j) + g_i(X_i)$;

3 Compute residuals $\hat{\varepsilon}_i = X_i - \hat{X}_i$ and $\hat{\varepsilon}_j = X_j - \hat{X}_j$;

4 **if** $\hat{\varepsilon}_i \not\perp\!\!\!\perp \{pa(i) \cup j\}$ *and* $\hat{\varepsilon}_j \not\perp\!\!\!\perp \{pa(j) \cup i\}$ **then**

5 Orient (i, j) as $i \leftrightarrow j$;

6 **if** $i \circ\!\!\rightarrow j$ *and* $\hat{\varepsilon}_j \perp\!\!\!\perp \{pa(j) \cup i\}$ **then**

7 Orient (i, j) as $i \rightarrow j$;

8 **else if** $i \circ\!\!-\!\!\circ j$ **then**

9 $(i, j) \leftarrow$ LikelihoodTest$\{\mathcal{G}, X, (i, j), \alpha\}$ (Algorithm 4).

After evaluating the subset U_{CI}, the algorithm proceeds to evaluate the remaining undetermined edges on Lines 16-20. These are edges for which the graphical criterion which dependencies detected through conditional independence tests cannot be correctly attributed to a causal relation or latent confounder. To ensure more reliable orientation, the algorithm first considers the subset of edges with the fewest shared neighbors (Line 18), since they have less possible common parents, which may cause latent confounding if not recovered. Each selected edge is then oriented sequentially using the score-based OrientByScore procedure, described in Algorithm 7. Given the identified parents and spouses of nodes i and j in the current graph $\mathcal{G}$,

OrientByScore evaluates possible edge orientation by comparing the resulting bivariate conditional log-likelihood $\log p(X_i, X_j \mid X_{-ij})$ (Line 20). The procedure selects the orientation that maximizes this bivariate log-likelihood. The full details of this procedure are expounded in Section 4.5.

4.5 Orientation by Score-Based Method

As demonstrated in Section 4.4.1, certain undetermined edges cannot be oriented using conditional independence tests because we cannot correctly attribute the residual dependence to a latent confounder. The graphical criterion in Proposition 2 can capture such edges, which require additional structure beyond CI tests. To address these challenges, we develop a score-based orientation framework. In the most general case, an undetermined edge $i \circ\!\!-\!\!\circ j$ admits three possible directionalities: $i \to j, j \to i, i \leftrightarrow j$. Different candidate orientations induce different parent and spouse sets, which in turn alter both the nonlinear regression functions and the residual covariance structure. Under the correction orientation, the conditional covariance between error terms should be more consistent with the implied graph structure and yield a greater likelihood. The central idea then is to compare the bivariate conditional log-likelihood of X_i, X_j given their parent and spouse nodes and orient the edge in the direction that maximizes the log-likelihood. To this end, we develop a robust likelihood-based estimation framework for both directed and bidirected edge structures.

4.5.1 Estimation of Log-Likelihood

In a mixed graph $\mathcal{G}$, we assume the observed variables $\{X_i\}_{i=1}^m$ follow a CAM (1.7),

$$X_i = \sum_{k \in \mathrm{pa}_{\mathcal{G}}(i)} f_{ik}(X_k) + \varepsilon_i,$$

where $\{\varepsilon_i\}_{i=1}^m \sim N(0, \Omega)$. For distinct nodes $i \neq j$, the off-diagonal entry $\omega_{ij} = \Omega_{ij}$ is characterized by the bidirected edge structure of $\mathcal{G}$ where

$$\omega_{ij} \neq 0 \Longleftrightarrow i \leftrightarrow j \text{ in } \mathcal{G} \tag{4.4}$$

The relation $\varepsilon_i = X_i - \sum_{k \in \mathrm{pa}_{\mathcal{G}}(i)} f_{ik}(X_k) = g(X_i)$ defines a transformation between ε and X. Through the change-of-variables formula, we have

$$p_X(X) = p_\varepsilon(g(X))|det(J_g(X))|,$$

where $J_g(X) = \frac{\partial g(X)}{\partial X}$ is the Jacobian matrix of the transformation. Since $\mathcal{G}$ is acyclic and $\{X_i\}_{i=1}^m$ follows an ANM, this implies $det(J_g(X)) = 1$.

We focus on maximizing the joint log-likelihood over the error terms, which is given as

$$\log p(\varepsilon_1, \ldots, \varepsilon_p \mid f, \Omega) = -\frac{N}{2} \log det(\Omega) - \frac{1}{2} tr[\Omega^{-1} \varepsilon \varepsilon^T]. \tag{4.5}$$

Consequently, the log-likelihood is parameterized by the set of nonlinear structural functions f, which determines the directed components of the graph, and the error covariance matrix Ω, which encodes the confounding structure through the bidirected edges.

We search for the edge direction (i, j) that maximizes the log-likelihood function. By the decomposition of the joint log-likelihood as

$$\log p(\varepsilon \mid f, \Omega) = \log p(\varepsilon_i, \varepsilon_j \mid \varepsilon_{-ij}) + \log p(\varepsilon_{-ij}),$$

we can treat the remaining nodes ε_{-ij} as fixed and restrict our attention to maximizing $p(\varepsilon_i, \varepsilon_j \mid \varepsilon_{-ij})$, since the remaining graph is held fixed across different orientations of (i, j).

Let C denote the block covariance matrix

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} \Omega_{ij,ij} & \Omega_{ij,-ij} \\ \Omega_{-ij,ij} & \Omega_{-ij,-ij} \end{bmatrix}$$

for which $C_{11} = \Omega_{ij,ij}$, $C_{12} = \Omega_{ij,-ij}$, and $C_{22} = \Omega_{-ij,-ij}$.

The bivariate conditional density $p(\varepsilon_i, \varepsilon_j \mid \varepsilon_{-ij})$ follows a joint normal distribution with mean μ_{ij} and variance Σ defined as

$$\begin{aligned} \mu_{ij} &= \mathbb{E}(\varepsilon_{ij} \mid \varepsilon_{-ij}) = C_{12} C_{22}^{-1} \varepsilon_{-ij}, \\ \Sigma_{ij} &= \mathbb{V}\mathrm{ar}(\varepsilon_{ij} \mid \varepsilon_{-ij}) = C_{11} - C_{12} C_{22}^{-1} C_{21}. \end{aligned} \tag{4.6}$$

4.5.2 Algorithm for Edge Orientation

Our score-based framework aims to find the edge orientation of (i, j) that maximizes $\log p(\varepsilon \mid f, \Omega)$, with other error terms and their associated parameters held constant. To obtain the parameters estimates f, Ω for the log-likelihood function over graphs with different orientations of (i, j), we use the Residual Iterative Conditional Fitting (RICF) algorithm proposed in Drton et al. [2009].

Specifically, we can apply the algorithm by fixing $p(\varepsilon_{-i})$ and update parameters from optimizing over $p(\varepsilon_i \mid \varepsilon_{-i})$. Let $\omega_{ii|-i}$ denote the conditional covariance of ε_i given ε_{-i} and $Z_h = (\Omega^{-1}_{-i,-i}\varepsilon_{-i})_h$ be a pseudo-spouse node of i, where $\omega_{i,h} \neq 0$. The

decomposition can be expressed as

$$\log p(\varepsilon \mid f, \Omega) = -\frac{N}{2} \log \omega_{ii|-i} - \frac{1}{2\omega_{ii|-i}} \left\| X_i - \sum_{k \in \mathrm{pa}_{\mathcal{G}}(i)} f_{ik}(X_k) - \sum_{h \in \mathrm{sp}_{\mathcal{G}}(i)} \omega_{ih} Z_h \right\|^2 - \frac{N}{2} \log \det(\Omega_{-i,-i}) - \frac{1}{2} \operatorname{tr}\left(\Omega_{-i,-i}^{-1} \varepsilon_{-i} \varepsilon_{-i}^T\right). \tag{4.7}$$

On each iteration, the RICF algorithm performs a node-wise optimization a variable ε_{-i} by fixing the marginal distribution of ε_{-i} and fitting the conditional distribution $p(\varepsilon_i \mid \varepsilon_{-i})$, then updating Ω. The effect of spouses are captured by Z_h for the algorithm, thus enabling log-likelihood estimation of mixed graphs. As the remaining variables ε_{-i} are fixed, maximizing the joint log-likelihood is equivalent to maximizing the first two terms of (4.7), which is the conditional log-likelihood $\log p(\varepsilon_i \mid \varepsilon_{-i})$. This yields updated estimates for f, ω_{ih}, and $\omega_{ii|i}$; as such, we can estimate $\omega_{ii} = \omega_{ii|-i} + \Omega_{i,-i}\Omega_{-i,-i}^{-1}\Omega_{-i,i}$ using the Schur complement.

The full edge orientation procedure is described in Algorithm 7. The algorithm begins with estimating the log-likelihood on Line 1 under each possible direction of $(i, j) \in \{\circ\!\!\rightarrow, \circ\!\!-\!\!\circ\}$. Using the RICF algorithm, it obtains and updates parameter estimates associated with ε_i first on Line 3, then those of ε_j on Line 4. With the updated sample covariance matrix Ω, our procedure evaluates the log-likelihood function $p(\varepsilon_{ij} \mid \varepsilon_{-ij})$ (Line 5). We use sample-splitting in our implementation so that parameter estimates are independent of the data to evaluate the log-likelihood function. Thereafter, it selects the edge direction among the three possibilities with the highest log-likelihood value (Line 7) and orients the edge accordingly in $\mathcal{G}$. In the case where the edge has an invariant edge mark on Line 8, i.e. $i \circ\!\!\rightarrow j$, the algorithm chooses from the two possible edge directions.

Algorithm 7: Score-based edge orientation(OrientByScore)

Input: Graph $\mathcal{G} = (V, E)$, observed data $X = \{X_i\}_{i=1}^{p}$, edge (i, j), error tolerance δ

Output: Graph $\mathcal{G} = (V, E)$

for $(i, j) \in \{i \to j, j \to i, i \leftrightarrow j\}$ **do**
  while $\ell_t - \ell_{t-1} > \delta$ **do**
    Update parameter estimates of $f_{ik}, k \in \text{pa}(i)$, $\omega_{ih}, h \in \text{sp}(i)$, $\omega_{ii|-i}$, and ω_{ii} over $p(\varepsilon_i \mid \varepsilon_{-i})$ (4.7), holding $p(\varepsilon_{-i})$ fixed ;
    Update parameter estimates of $f_{jk}, k \in \text{pa}(j)$, $\omega_{jh}, h \in \text{sp}(j)$, $\omega_{jj|-j}$, and ω_{jj} over $p(\varepsilon_j \mid \varepsilon_{-j})$ (4.7), holding $p(\varepsilon_{-j})$ fixed ;
    Compute $\ell_t = \log p(\varepsilon_{ij} \mid \varepsilon_{-ij})$ (4.6);
if $i \circ\!\!-\!\!\circ j$ **then**
  Obtain the preferred edge orientation $E_{i,j} = \max[\ell(i \to j \mid X, \mathcal{G}), \ell(j \to i \mid X, \mathcal{G}), \ell(i \leftrightarrow j \mid X, \mathcal{G})]$;
else
  Obtain the preferred edge orientation $E_{i,j} = \max[\ell(i \to j \mid X, \mathcal{G}), \ell(i \leftrightarrow j \mid X, \mathcal{G})]$;
Orient (i, j) as $E_{i,j}$.

When assumptions for using conditional independence tests are not satisfied, we can determine edge orientations using this edge orientation procedure by optimizing for the log-likelihood. Moreover, the RICF algorithm allows us to compute the log-likelihood for mixed graphs and enables comparison across candidate graph structures.

4.6 Experiment Results

We have conducted a series of experiments on synthetic data to assess the algorithm's ability to (i) recover the latent representation over observed variables and (ii) distinguish causal relations from confounding effects. To estimate the initial graph, we implemented a modified version of the FCI algorithm Spirtes et al. [2000] using the R package **pcalg** [Kalisch et al., 2012], with the kernel-based HSIC test from **kpcalg** [Gretton et al., 2009] as the conditional independence test. In Algorithm 6, residual independence is assessed using mutual information as the dependence measure, after discretizing variables as a preprocessing step. For the likelihood ratio test, we adopt the sample-splitting variant provided in the R package **SNOE** [Huang and Zhou, 2025]. Unless otherwise specified, all statistical tests use a significance level of $\alpha = 0.05$; for the residual independence test, the significance level is adjusted under multiple testing. All regression models are implemented as generalized additive models (GAMs) using the **mgcv** package [Wood, 2001], with thin-plate splines as the basis functions. Practitioners may choose to conduct a pruning step as the final stage to remove extraneous edges, if necessary, where we compare the log-likelihood of the current edge orientation against the case with no edge between the nodes.

We compared our method against several ADMG and DAG learning algorithms

with similar model assumptions. CAM-UV Maeda and Shimizu [2021] is a constraint-based ADMG learning method that uses a kernel-based conditional independence test to evaluate the independence between residuals and increasingly larger subsets of covariates; when causal direction is unidentifiable, the pair of nodes are considered to have a latent confounder on an unobserved frontdoor or backdoor path between them. SNOE-SS and SNOE-CV are two variants of a DAG learning algorithm [Huang and Zhou, 2025] that assume a CAM (1.7) with additive Gaussian noise and recover the DAG by sequentially orienting edges in a CPDAG. CAM [Bühlmann et al., 2014] is a score-based DAG learning method under the CAM with additive Gaussian noise as well. It first estimates the skeleton and then optimizes the log-likelihood to recover a topological ordering. For all methods, we used the recommended parameter settings. To measure the incremental contribution of our edge orientation procedure, we additionally report its performance when provided with the ground-truth PAG as input.

In all experiments, the ground-truth graph is an ADMG constructed as the latent projection of modified DAG structures, ranging from 8 to 44 nodes in the original structure, from the **bnlearn** network repository. We evaluate the methods on ADMGs with varying percentages of bidirected edges. For each DAG structure, we calculate the number of bidirected edges needed to achieve varying percentages of bidirected edges in the graph and add a latent common cause to unique pairs of nodes accordingly. In essence, the new DAG structure only differs in containing bidirected edges between certain non-adjacent nodes. For each latent confounding level and ADMG structure, we generated $N = 150$ datasets with $n = 2000$ samples under a CAM (1.7). The covariate functions f_{ij} are randomly selected from three distinct function classes: (i) invertible functions (piecewise, exponential, cubic, and inverse sine functions), (2)

non-invertible functions (piecewise, quadratic, cubic, and inverse tangent functions), and (3) functions drawn from Gaussian processes using a squared exponential kernel and bandwidth $l \sim Unif[5, 5.25]$. The error terms are generated independently as $\varepsilon_i \sim N(0, \sigma_i^2)$, where $\sigma_i \sim \text{Unif}(0.5, 0.75)$ for all $i = 1, \ldots, p$. Our primary evaluation metric is the F1 score defined as $F1 = \frac{2TP}{2TP+FP+FN+2IO}$, defined in Chapter 3.1.3.

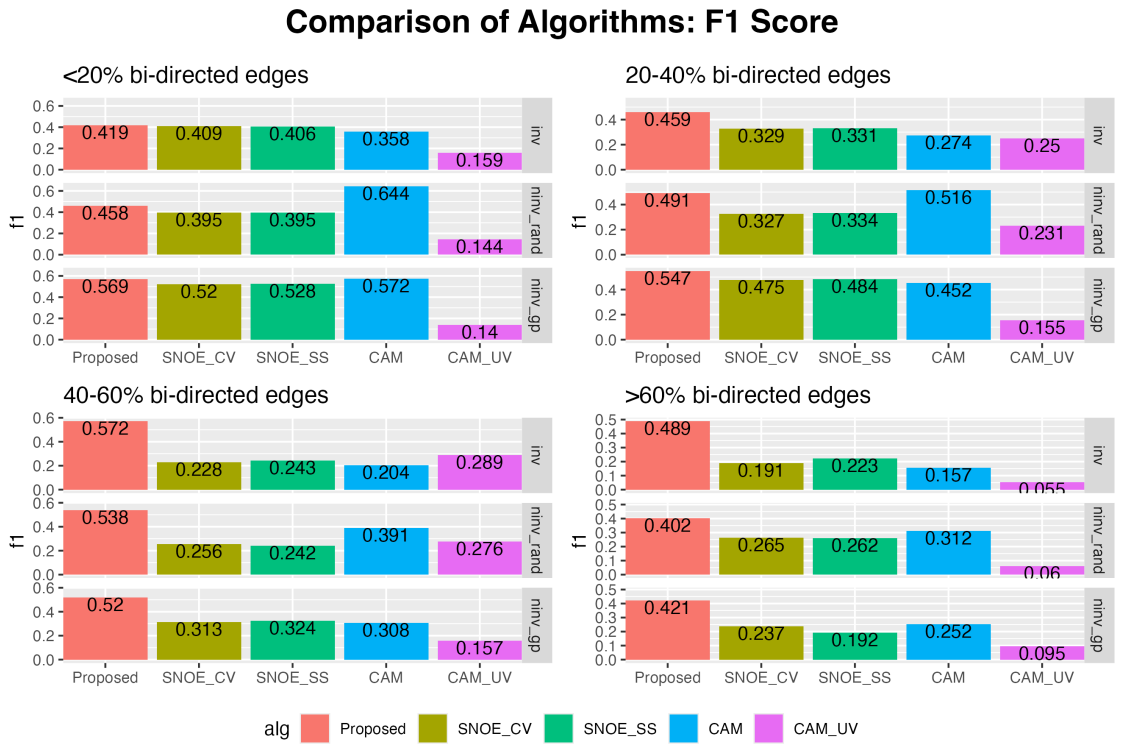


Figure 4.3: F1 Score of Initial Graph and After Edge Orientation

Across all experimental settings, our proposed algorithm demonstrates the strongest overall performance across the evaluated settings, seen in Figure 4.4, particularly as latent confounding increases. Under low latent confounding (<20% bidirected edges), it performs competitively against CAM and outperforms SNOE, with CAM occasionally matching or slightly exceeding performance under functions from Gaussian

processes. As the proportion of bidirected edges increases to moderate (20–40%), high (40–60%), and very high (>60%) confounding levels, the algorithm outperforms all competing methods, often by substantial margins. This trend is especially pronounced relative to DAG-based baselines (SNOE-CV, SNOE-SS, and CAM), whose performance deteriorates sharply as latent confounding grows, reflecting the negative impact of latent variables. CAM-UV, while designed for ADMG learning, consistently underperforms across all settings and often yields the lowest F1 scores; a close analysis shows that it has a high count of missing edges and incorrectly oriented edges.

A key advantage of our method is its empirical robustness. Whereas competing methods show declines as the percentage of bidirected edges increases, our method maintains comparatively high F1 scores even when more than 60% of edges are affected by latent variables. In several heavily confounded settings, it improves the F1 score by approximately 0.15-0.3. Even with few latent confounders in the ADMG, our algorithm demonstrates comparable accuracies with the DAG learning methods.

The intermediate performance analysis in Figure 4.3 shows a more nuanced pattern than a simple monotonic improvement in the graph accuracy. Given the true PAG, the sequential edge orientation uncovers more edge marks as the F1 score increases by 0.15-0.2 in all settings; the amount of improvement is also consistent across all classes of data generating functions, indicating its ability to learn from a broader class of functional forms. The orientation stage itself consistently improves ADMG recovery when starting from the estimated PAG: in every confounding regime and function class, the final graph substantially outperforms the initial graph, often by large margins. The improvements are particularly great in low and moderate confounding settings, where F1 increases by roughly 0.20–0.35 in several cases. These gains are especially pronounced when the initial PAG is exact, demonstrating that

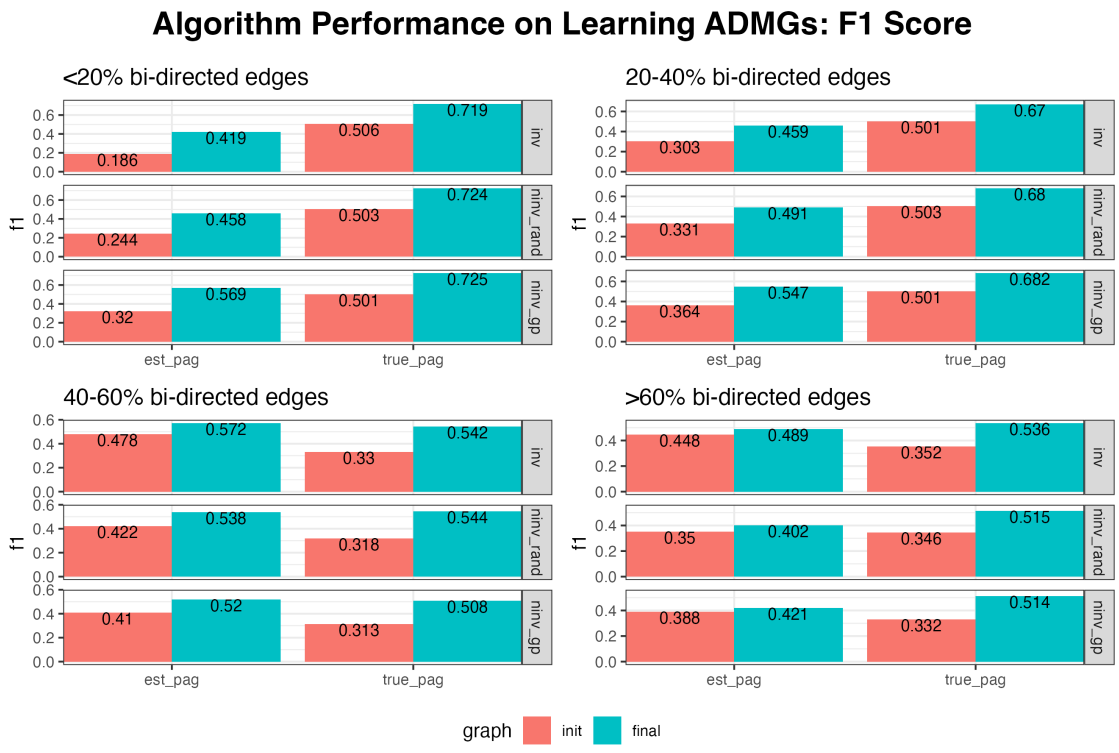


Figure 4.4: F1 Score of Competing Algorithms

the orientation procedure provides great improvement beyond the initial PAG output.

Overall, these results indicate that our edge orientation procedure is both competitive and scalable across a wide range of nonlinear causal discovery settings, with particular strengths in distinguishing causal from confounded relations under substantial latent structure. The method's consistent gains over both ADMG and DAG learning algorithms reflect its ability to capture and distinguish between causal and confounding relations.

4.7 Discussion

In this work, we propose a causal discovery framework for identifying variable relationships in the presence of latent confounding under nonlinear additive noise models. Given a PAG, our method sequentially orients edges with undetermined marks by combining a constraint-based strategy, which directly tests for confounding and causal relations, with a likelihood-based approach that leverages the covariance structure to directly compare competing edge orientations under a nonlinear SEM.

In particular, the proposed graphical criterion allows us to identify which edges in the graph can be readily oriented via conditional independence tests, which can save computational costs compared to global search and iterative testing frameworks. Our log-likelihood comparison framework enables direct comparison between mixed graphs under the nonlinear model. Moreover, the procedure only requires maximizing conditional edge-wise log-likelihoods rather than the likelihood of the entire graph. Simulation results show that the proposed orientation procedure substantially improves upon the initial graph, whether the initial PAG is exact or estimated, while outperforming competing algorithms under similar model assumptions. Together, these components enable more precise recovery of underlying causal relations beyond standard equivalence-class outputs.

CHAPTER 5

Conclusion and Discussion

In this dissertation, we develop several methodological advances for identifying nonlinear causal relationships in graphical models under both causal sufficiency and latent confounding. We first introduce a general framework for orienting undirected edges within the Markov equivalence class of a DAG through a sequential procedure, termed the Sequential Nonlinear Orientation of Edges (SNOE) in Chapter 2. We establish theoretical guarantees for this approach by showing that, using the CPDAG as the initial graph, there exists at least one edge whose orientation can be correctly identified at any stage (Section 2.2.2). This edge is characterized via pairwise additive noise model (PANM), a criterion we propose for local identifiability, which serves as the foundation of the algorithm.

The SNOE procedure described in Section 2.3 consists of two key components: a ranking mechanism and a likelihood ratio test. Leveraging the independent noise property entailed by the PANM, we construct a nonlinear independence measure to rank undirected edges, ensuring that edges satisfying the PANM criterion are oriented first. This addresses an issue common to constraint-based methods, where early orientation errors can propagate and degrade overall performance. We then employ a likelihood ratio test to determine the causal direction by comparing the factorization of the bivariate conditional density under competing orientations, $X \to Y$ and

$Y \to X$. We further show in Section 2.3.3 that, under the null hypothesis, the test statistic converges to a standard Normal distribution, which enables straightforward computation and interpretation. Empirical evaluations on both simulated and real-world datasets in Chapter 3 demonstrate that SNOE achieves high accuracy and robustness relative to existing approaches, while also exhibiting strong performance of its individual components in edge orientation.

In Chapter 4, we develop a causal discovery method for graphical models in the presence of latent confounders by the sequential orientation of edges in a PAG. Unlike methods that rely on exhaustive testing or global search over graph structures, we introduce a graphical criterion (stated in Proposition (2)) to identify edges for which conditional independence test results can be reliably interpreted. Specifically, edges satisfying this criterion allow residual dependence to be more reliably attributed to latent confounding rather than unidentified observed parents. For other edges, we propose a likelihood-based approach in Chapter 4.5. We utilize an iterative parameter estimation procedure to recover both the structural functions in the SEM and the covariance matrix that encodes the presence of bidirected edges. These estimates are then used to compute and compare log-likelihoods under possible edge orientations. This hybrid approach effectively integrates the strengths of constraint-based methods—namely, the ability to target specific edges— with those of score-based approaches, which facilitate direct comparison between graphical structures. Empirical results presented in Section 4.6 demonstrate that the proposed method yields substantial improvements over the initial PAG and shows higher accuracy compared to competing methods.

As such, both frameworks have several limitations that suggest directions for further improvement. First, the practical methodology for both works is developed

under the causal additive model, which, while flexible, still imposes structural restrictions on the data-generating process. Extending the approach to a more general nonlinear additive noise model (1.5)—such as models with non-additive interactions among covariates or non-Gaussian noise distributions— would substantially broaden its applicability. The present implementation relies on relatively structured regression techniques, which may limit performance when the underlying relationships are highly nonlinear or high-dimensional. Incorporating more flexible functions, such as neural networks, offers a promising direction for improving model estimation [Uemura and Shimizu, 2020]. Another direction is to relax model assumptions under the hidden variable setting to varying numbers of children per latent confounder.

Second, while the proposed graphical criterion underlying the latent-variable orientation procedure allows for more systematic edge orientation, this criterion may not yet fully exploit all patterns present in the graph. Further refinement and expansion of these graphical conditions could increase the number of edges that can be reliably oriented, thereby improving overall recovery of the causal graph, such as methods in Li et al. [2025] and Ling et al. [2025] that uncover edge orientations to achieve a more informative PAG.

Finally, a more comprehensive theoretical and empirical analysis of the latent-variable orientation procedure is needed. In particular, quantifying its soundness and completeness in terms of exact precision and recall for edge mark recovery would provide a deeper understanding of its reliability and limitations. Strengthening these theoretical guarantees can also guide practical use of the method in finite-sample settings. An empirical analysis of the algorithm under certain scenarios, such as latent confounders on non-unique pairs and non-separable effects of observed and latent nodes, provides insights into its performance under violations of model

assumptions. Moreover, developing sharper identifiability assumptions and results for causal relations under latent confounders with an underlying nonlinear SEM remains an important open problem.

APPENDIX A

Additional Results

A.1 Likelihood Ratio under Gaussian Regression Models

In this section, we verify conditions (i) - (iii) of Theorem 2 under the Gaussian Regression model:

$$X = m(Z) + \varepsilon, \qquad \varepsilon \sim N(0, \sigma^2), \tag{A.1}$$

where $Z = (Z_1, \ldots, Z_d)$, the noise ε is independent of Z, and $0 < \sigma^2 < \infty$. Suppose we compare two Gaussian regression models, $F(X \mid Z)$ and $G(X \mid Z)$. Without loss of generality, assume $F : X = m_F(Z) + \varepsilon_F$ is the true model, while $G : X = m_G(Z) + \varepsilon_G$ is not necessarily true. In particular, properties of the error variable, $\varepsilon_F \sim N(0, \sigma_F^2)$ and $\varepsilon_F \perp\!\!\!\perp Z$, hold under model F, whereas they may not hold for ε_G if $m_G \neq m_F$.

Under the Gaussian regression model (A.1), the conditional densities $F(\cdot \mid Z)$ and $G(\cdot \mid Z)$ are of the form

$$F(x \mid z) = \frac{1}{\sqrt{2\pi\sigma_F^2}} \exp\left\{ -\frac{(x - m_F(z))^2}{2\sigma_F^2} \right\}, \qquad G(x \mid z) = \frac{1}{\sqrt{2\pi\sigma_G^2}} \exp\left\{ -\frac{(x - m_G(z))^2}{2\sigma_G^2} \right\},$$

where the noise variances σ_F^2, σ_G^2 are finite and positive. Let $\widehat{m}_F, \widehat{m}_G$ and $\widehat{\sigma}_F^2, \widehat{\sigma}_G^2$ be estimators constructed from a training sample of size n. Denote by $\widehat{F}$ (and $\widehat{G}$) the corresponding estimator of the conditional density F (and G) after plugging in $\widehat{m}_F, \widehat{\sigma}_F^2$ (and $\widehat{m}_G, \widehat{\sigma}_G^2$). Let (X, Z) be a test sample independent of the training

sample.

A.1.1 Verification of Condition (i)

We start with

$$\log \frac{\widehat{F}(X \mid Z)}{\widehat{G}(X \mid Z)} = \frac{1}{2} \log \frac{\widehat{\sigma}_G^2}{\widehat{\sigma}_F^2} - \frac{(X - \widehat{m}_F(Z))^2}{2\widehat{\sigma}_F^2} + \frac{(X - \widehat{m}_G(Z))^2}{2\widehat{\sigma}_G^2}.$$

For any $r \geq 1$, there exists a constant $C_1 < \infty$ such that

$$\left| \log \frac{\widehat{F}(X \mid Z)}{\widehat{G}(X \mid Z)} \right|^r \leq C_1 \left[1 + \frac{(X - \widehat{m}_F(Z))^{2r}}{(\widehat{\sigma}_F^2)^r} + \frac{(X - \widehat{m}_G(Z))^{2r}}{(\widehat{\sigma}_G^2)^r} \right], \tag{A.2}$$

assuming $\widehat{\sigma}_F^2, \widehat{\sigma}_G^2$ are finite and bounded away from 0. Note that we can decompose the regression error $X - \widehat{m}_G(Z)$ as

$$X - \widehat{m}_G(Z) = \varepsilon_G + (m_G(Z) - \widehat{m}_G(Z))$$

to obtain a bound

$$|X - m_G(Z)|^{2r} \leq C_2 \left[|\varepsilon_G|^{2r} + |m_G(Z) - \widehat{m}_G(Z)|^{2r} \right],$$

where $C_2 > 0$, and a similar bound for $|X - m_F(Z)|^{2r}$. Let $r = 2 + \eta$ for some $\eta > 0$ and assume the following conditions:

(i-1) The error variables ε_F and ε_G have finite $(4 + 2\eta)$-th moments. Note that the Gaussian error ε_F satisfies this automatically.

(i-2) The regression estimators satisfy the uniform higher-order moment bound

$$\sup_n \mathbb{E}\Big[\big| \widehat{m}_F(Z) - m_F(Z) \big|^{4+2\eta} + \big| \widehat{m}_G(Z) - m_G(Z) \big|^{4+2\eta} \Big] < \infty.$$

(i-3) For all n, the variance estimators $\widehat{\sigma}_F^2, \widehat{\sigma}_G^2 \in [c, M]$ for some $0 < c < M < \infty$.

Then, the expectation of the right-hand side of (A.2) is bounded uniformly in n. Consequently,

$$\sup_n \mathbb{E}\left[\left|\log \frac{\widehat{F}(X \mid Z)}{\widehat{G}(X \mid Z)}\right|^{2+\eta}\right] < \infty,$$

which verifies Condition (i) of Theorem 2. We summarize this result as a lemma:

Lemma 3. *Under assumptions (i-1) to (i-3) in Appendix A.1.1, Condition (i) of Theorem 2 holds for the Gaussian regression models F and G.*

Remark 6. Assumptions (i-1) and (i-2) only require finite $(4 + 2\eta)$-th moments for ε_G and for the error of the regression function estimators. Assumption (i-3) is satisfied if the variance estimators are finite and bounded away from 0. These are all mild and easily satisfied for Gaussian regression models.

A.1.2 Verification of Condition (ii)

Define the population and estimated log-likelihood ratios as

$$R = \log \frac{F(X \mid Z)}{G(X \mid Z)} = \frac{1}{2}\log \frac{\widehat{\sigma}_G^2}{\widehat{\sigma}_F^2} - \frac{(X - \widehat{m}_F(Z))^2}{2\widehat{\sigma}_F^2} + \frac{(X - \widehat{m}_G(Z))^2}{2\widehat{\sigma}_G^2},$$
$$R_0 = \log \frac{\widehat{F}(X \mid Z)}{\widehat{G}(X \mid Z)} = \frac{1}{2}\log \frac{\sigma_G^2}{\sigma_F^2} - \frac{(X - m_F(Z))^2}{2\sigma_F^2} + \frac{(X - m_G(Z))^2}{2\sigma_G^2}.$$

We show that the following assumptions are sufficient for condition (ii):

(ii-1) $\mathbb{E}(\varepsilon_F^4), \mathbb{E}(\varepsilon_G^4) < \infty$,

(ii-2) $\widehat{m}_F(Z) \xrightarrow{L^2} m_F(Z), \widehat{m}_G(Z) \xrightarrow{L^2} m_G(Z)$,

(ii-3) $\widehat{\sigma}_F^2 \xrightarrow{L^2} \sigma_F^2, \widehat{\sigma}_G^2 \xrightarrow{L^2} \sigma_G^2$.

Assumptions (ii-2) and (ii-3) imply that each term in $R - R_0$ converges to zero in L^2, and consequently,

$$\mathbb{E}\big[(R - R_0)^2\big] \to 0. \tag{A.3}$$

Since ε_F is Gaussian and $\mathbb{E}(\varepsilon_G^4) < \infty$ by (ii-1), $\sigma_0^2 := \mathbb{V}\mathrm{ar}(R_0) < \infty$, and (A.3) implies $\mathbb{E}[R] \to \mathbb{E}[R_0]$ and $\mathbb{E}[R^2] \to \mathbb{E}[R_0^2]$, it follows that $\mathbb{V}\mathrm{ar}(R) \to \mathbb{V}\mathrm{ar}(R_0) = \sigma_0^2$. Let

$$U_n^2 := \mathbb{E}\Big[(R - R_0)^2 \mid \widehat{F}, \widehat{G}\Big].$$

By the tower property, $\mathbb{E}[U_n^2] = \mathbb{E}[(R - R_0)^2] \to 0$, which implies

$$\mathbb{E}\Big[(R - R_0)^2 \mid \widehat{F}, \widehat{G}\Big] = U_n^2 \xrightarrow{p} 0. \tag{A.4}$$

Consider the decomposition

$$\mathbb{V}\mathrm{ar}(R \mid \widehat{F}, \widehat{G}) = \mathbb{V}\mathrm{ar}(R_0 \mid \widehat{F}, \widehat{G}) + \mathbb{V}\mathrm{ar}(R - R_0 \mid \widehat{F}, \widehat{G}) + 2\,\mathbb{C}\mathrm{ov}(R_0, R - R_0 \mid \widehat{F}, \widehat{G}),$$

where $\mathbb{V}\mathrm{ar}(R_0 \mid \widehat{F}, \widehat{G}) = \sigma_0^2$ as R_0 does not depend on $(\widehat{F}, \widehat{G})$. Note that

$$\mathbb{V}\mathrm{ar}(R - R_0 \mid \widehat{F}, \widehat{G}) \le \mathbb{E}[(R - R_0)^2 \mid \widehat{F}, \widehat{G}]$$

and by the Cauchy–Schwarz inequality,

$$\Big|\mathbb{C}\mathrm{ov}(R_0, R - R_0 \mid \widehat{F}, \widehat{G})\Big| \le \sqrt{\mathbb{V}\mathrm{ar}(R_0)}\,\sqrt{\mathbb{E}[(R - R_0)^2 \mid \widehat{F}, \widehat{G}]}.$$

Combining these bounds with (A.4), we obtain $\mathbb{V}\mathrm{ar}(R \mid \widehat{F}, \widehat{G}) \xrightarrow{p} \sigma_0^2$, which establishes Condition (ii). The result is summarized as a lemma:

Lemma 4. *Under assumptions (ii-1) to (ii-3) in Appendix A.1.2, Condition (ii) of Theorem 2 holds for the Gaussian regression models F and G.*

Remark 7. In Gaussian additive models, Condition (ii) is implied by the L^2 consistency of the regression function estimators and the variance estimators, together with a mild 4-th moment condition for the error distributions. These conditions are standard for spline or series estimators under sample splitting.

A.1.3 Verification of Condition (iii)

Since F and G both define a Gaussian distribution for $[X \mid Z]$, we calculate $D(p\|\widehat{p})$ between

$$p(x \mid z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x - m(z))^2}{2\sigma^2}\right\} \quad \text{and} \quad \widehat{p}(x \mid z) = \frac{1}{\sqrt{2\pi\widehat{\sigma}^2}} \exp\left\{-\frac{(x - \widehat{m}(z))^2}{2\widehat{\sigma}^2}\right\}.$$

Let $\varepsilon := X - m(Z)$. Given $(\widehat{m}, \widehat{\sigma}^2)$, taking expectation with respect to (X, Z), we have

$$D(p\|\widehat{p}) = \frac{1}{2}\left[\log\left(\frac{\widehat{\sigma}^2}{\sigma^2}\right) - 1\right] + \frac{\mathbb{E}[(X - \widehat{m}(Z))^2 \mid \widehat{m}]}{2\widehat{\sigma}^2},$$

using that $\mathbb{E}(X - m(Z))^2 = \mathbb{V}\mathrm{ar}(\varepsilon) = \sigma^2$ for both F and G. Note that

$$(X - \widehat{m}(Z))^2 = \varepsilon^2 + (m(Z) - \widehat{m}(Z))^2 + 2\varepsilon(m(Z) - \widehat{m}(Z)). \tag{A.5}$$

Since F is the true model, $\varepsilon_F \perp\!\!\!\perp Z$ and thus $\mathbb{E}[\varepsilon_F(m_F(Z) - \widehat{m}_F(Z)) \mid \widehat{m}_F] = 0$. Taking conditional expectation of (A.5) given $\widehat{m}_F$, we have

$$D(F\|\widehat{F}) = \frac{1}{2}\left[\log\left(\frac{\widehat{\sigma}_F^2}{\sigma^2}\right) + \frac{\sigma^2}{\widehat{\sigma}_F^2} - 1\right] + \frac{\mathbb{E}[(\widehat{m}_F(Z) - m_F(Z))^2 \mid \widehat{m}_F]}{2\widehat{\sigma}_F^2}. \tag{A.6}$$

To achieve the required rates in condition (iii), we assume the following:

(iii-1) $\mathbb{E}(\widehat{\sigma}^2 - \sigma^2)^2 = o(n^{-1/2})$ for both $\widehat{F}$ and $\widehat{G}$.

(iii-2) $\mathbb{E}(\widehat{m}(Z) - m(Z))^2 = o(n^{-1/2})$ for both $\widehat{F}$ and $\widehat{G}$.

The function $g(t) = \log(t/\sigma^2) + \sigma^2/t - 1$ satisfies $g(\sigma^2) = g'(\sigma^2) = 0$ and $g''(\sigma^2) = 1/\sigma^4$, hence a Taylor expansion shows that $g(\widehat{\sigma}^2) = O\big((\widehat{\sigma}^2 - \sigma^2)^2\big)$. Consequently, if $\mathbb{E}(\widehat{\sigma}^2 - \sigma^2)^2 = o(n^{-1/2})$, then the first term, $\frac{1}{2}g(\widehat{\sigma}_F^2)$, on the right-hand side of (A.6) is $o_p(n^{-1/2})$. By Assumption (iii-1), $\widehat{\sigma}^2 \to \sigma^2$ in probability and is bounded away from

zero. The second term in (A.6) is then $o_p(n^{-1/2})$ due to the L_2 rate in (iii-2). This shows that condition (iii) with $b = 1/2$ is satisfied for $\widehat{F}$.

For $D(G\|\widehat{G})$, the only difference is that we also need to consider the expectation of the last term in (A.5) since ε_G is not necessarily independent of Z. By the Cauchy-Schwarz inequality,

$$|\mathbb{E}[\varepsilon_G(m_G(Z) - \widehat{m}_G(Z)) \mid \widehat{m}_G]| \leq [\mathbb{V}\mathrm{ar}(\varepsilon_G)]^{1/2} \left\{\mathbb{E}[(\widehat{m}_G(Z) - m_G(Z))^2 \mid \widehat{m}_G]\right\}^{1/2}.$$

Assumption (iii-2) implies that $\mathbb{E}[(\widehat{m}_G(Z) - m_G(Z))^2 \mid \widehat{m}_G] = o_p(n^{-1/2})$ and thus

$$\mathbb{E}[\varepsilon_G(m_G(Z) - \widehat{m}_G(Z)) \mid \widehat{m}_G] = o_p(n^{-1/4}).$$

Together with $\widehat{\sigma}_G^2 \xrightarrow{p} \sigma_G^2 > 0$ due to (iii-1), this shows that $D(G\|\widehat{G}) = o_p(1)$, i.e., condition (iii) holds with $b = 0$ for $\widehat{G}$.

We summarize these results into a lemma:

Lemma 5. *Under assumptions (iii-1) and (iii-2) in Appendix A.1.3, Condition (iii) of Theorem 2 holds for $\widehat{F}$ with $b = 1/2$ and holds for $\widehat{G}$ with $b = 0$.*

A.2 Proofs

A.2.1 Proof of Theorem 1

Before proving Theorem 1, we first state a relevant result:

Lemma 6. *Suppose the variables (nodes) of the DAG $\mathcal{G}_0$ follow a restricted ANM. Let (X, Y) be an undirected edge in $\mathcal{G}$ on Line 2 at any stage of Algorithm 1 with the input being the CPDAG of $\mathcal{G}_0$. Let $Z_1 = pa_{\mathcal{G}}(X)$ and $Z_2 = pa_{\mathcal{G}}(Y)$. If $pa_{\mathcal{G}_0}(X) = Z_1, pa_{\mathcal{G}_0}(Y) = Z_2 \cup \{X\}$ or $pa_{\mathcal{G}_0}(Y) = Z_2, pa_{\mathcal{G}_0}(X) = Z_1 \cup \{Y\}$, then $[X, Y|Z_1, Z_2]$ follows a pairwise additive noise model.*

Lemma 6 identifies a type of undirected edge, amongst all, that satisfy the PANM. We will show that the PDAG $\mathcal{G}$ contains at least one such undirected edge on Line 2 at any iteration of the algorithm. Now we prove the theorem:

Proof. To prove that the sequential edge orientation procedure can recover the true DAG $\mathcal{G}_0$ by correctly orienting all undirected edges in the CPDAG $\mathcal{E}$, we demonstrate that the following holds true for $\mathcal{G}$ at every iteration at Line 2 in Algorithm 1: If $\mathcal{G}$ is not a DAG, then there exists an undirected edge $X - Y$ such that $[X, Y \mid Z_1, Z_2]$ satisfies the PANM, where $Z_1 = \text{pa}_{\mathcal{G}}(X)$ and $Z_2 = \text{pa}_{\mathcal{G}}(Y)$. It is easy to see that every orientation step in Algorithm 1 will only lead to correct orientation that is consistent with the DAG $\mathcal{G}_0$ if the input $\mathcal{G}$ is the CPDAG.

Let T be an undirected component of the PDAG $\mathcal{G}$ of size $|T| \geq 2$. Let $v_1 \in T$ be a node that precedes all other nodes in T according to some topological ordering $\prec$ of $\mathcal{G}_0$. Because $|T| \geq 2$, the neighbor set $\text{ne}_{\mathcal{G}}(v_1)$ is not empty. If there exists $v_2 \in \text{ne}_{\mathcal{G}}(v_1)$ such that $\{v_1, v_2\}$ satisfies the conditions in Lemma 6, then (v_1, v_2) satisfies the PANM and the proof is complete. By construction all parents of v_1 have been identified in $\mathcal{G}$, i.e. $\text{pa}_{\mathcal{G}}(v_1) = \text{pa}_{\mathcal{G}_0}(v_1)$. Let v_2 be a node that precedes all other nodes in $\text{ne}_{\mathcal{G}}(v_1)$ according to some sort of $\mathcal{G}_0$. Then, $v_1 \in \text{pa}_{\mathcal{G}_0}(v_2)$ and none of the nodes in $\text{ne}_{\mathcal{G}}(v_1)$ is a parent of v_2 in $\mathcal{G}_0$. It remains to show that $\text{pa}_{\mathcal{G}_0}(v_2) = \text{pa}_{\mathcal{G}}(v_2) \cup \{v_1\}$. If this is not the case, then there must exist another node $v_j \in \text{ne}_{\mathcal{G}}(v_2) \setminus \{v_1\}$ that is a parent of v_2 in $\mathcal{G}_0$ and $v_j \notin \text{ne}_{\mathcal{G}}(v_1)$, i.e. there is no undirected edge between v_1 and v_j in the PDAG $\mathcal{G}$. There are two possibilities with respect to the connectivity between v_1 and v_j in $\mathcal{G}_0$. The first possibility is that there is no edge between v_1 and v_j in $\mathcal{G}_0$. This would form a new v-structure $v_1 \to v_2 \leftarrow v_j$ in $\mathcal{G}_0$, which is a contradiction to that the input $\mathcal{G}$ is the CPDAG of $\mathcal{G}_0$. The second possibility is that $v_1 \to v_j$ in $\mathcal{G}_0$

since $v_1 \prec v_j \in T$ and this edge has been oriented so in $\mathcal{G}$. Since neither $v_1 - v_2$ nor $v_2 - v_j$ has been oriented in $\mathcal{G}$, the edge $v_1 \to v_j$ must have been oriented either by Line 7 or by Meek's rule 1 on Line 8. In what follows, we show that both scenarios would lead to contradictions, and thus such v_j does not exist.

If $v_1 \to v_j$ is oriented by Line 7, then there must be another node v_i such that $v_1 - v_i$ was oriented in either direction on Line 4 first and v_i is adjacent to v_j. The algorithm would also orient $v_i \to v_j$ by Line 7 or from previous actions. Shown in Figure A.1 (a) and (b), there are two cases regarding the adjacency of v_2 and v_i in $\mathcal{G}$ assuming $v_1 \to v_i$ has been oriented. (a) If v_i, v_2 are adjacent, v_i must be a parent of v_2 because otherwise there would be a directed cycle $v_2 \to v_i \to v_j \to v_2$ in $\mathcal{G}_0$. Then, Line 7 would orient $v_1 \to v_2$ too, contradicting to that $v_1 - v_2$ is undirected in $\mathcal{G}$. (b) If v_i, v_2 are not adjacent, then the algorithm would orient $v_j \to v_2$ by Meek's rule 1 in the following step, contradicting to that $v_j \in \text{ne}_{\mathcal{G}}(v_2)$ (i.e. $v_j - v_2$ in $\mathcal{G}$). Similar arguments under the orientation $v_i \to v_1$ result in contradictions that supposedly undirected edges in $\mathcal{G}$ would have been oriented.

If $v_1 \to v_j$ is oriented by Meek's rule 1 on Line 8, then there must exist a node $Z \in V$ in the configuration $Z \to v_1 - v_j$ and Z is not adjacent to v_j. There are four possible cases, depicted in Figure A.2, with respect to the connection between Z and v_2 in the PDAG $\mathcal{G}$. Case 1: there is no edge between Z and v_2. The undirected edge $v_1 - v_2$ would then be oriented by rule 1 as $v_1 \to v_2$, which leads to a contradiction. Case 2: The two nodes are connected by an undirected edge $Z - v_2$. Then, $Z \in T$ and is a parent node of v_1, contradicting the fact that v_1 precedes all other nodes in T. Case 3: $Z \to v_2$ in $\mathcal{G}$. Meek's rule 1 would then orient $v_2 \to v_j$, which would result in an incorrect orientation as v_j is assumed to be a parent of v_2 in $\mathcal{G}_0$, again a contradiction. Case 4: $v_2 \to Z$ in $\mathcal{G}$. Then the edge $v_1 - v_2$ must be $v_2 \to v_1$ in $\mathcal{G}_0$,

(a) v_i, v_2 are adjacent (b) v_i, v_2 are not adjacent

Figure A.1: First possibility: orienting $v_1 \to v_j$ by Line 7 in Algorithm 1 results in contradictions where $v_1 \to v_2$ is oriented in (a) or $v_j \to v_2$ in (b).

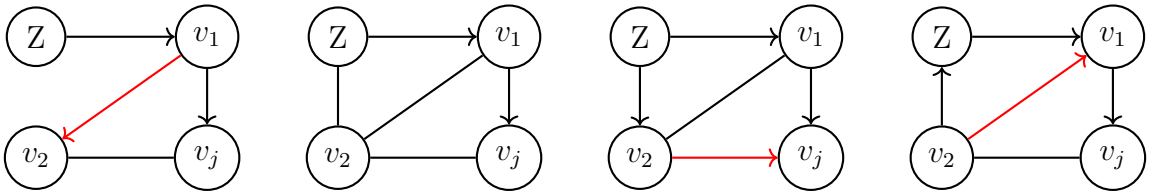


Figure A.2: Second possibility: edge $v_1 \to v_j$ can be oriented by Meek's rule 1 under four cases, all of which lead to a contradiction.

which contradicts the assumed ordering $v_1 \prec v_2$. □

A.2.2 Proof of Theorem 2

Proof. Conditioning on $(\widehat{F}, \widehat{G})$, the variables $\{R_i\}_{i=1}^n$ are i.i.d. with conditional mean

$$m_n = \mathbb{E}\left[\log \frac{\widehat{F}(X \mid Z)}{\widehat{G}(X \mid Z)} \,\middle|\, \widehat{F}, \widehat{G}\right]$$

and conditional variance v_n. By Lyapunov central limit theorem, applicable because of condition (i),

$$\Big[\frac{\sqrt{n}(\bar{R}-m_n)}{\sqrt{v_n}} \;\Big|\; (\widehat{F},\widehat{G})\Big] \xrightarrow{D} N(0,1). \tag{A.7}$$

It is easy to see that

$$\mu_0 - m_n = D(F \,\|\, \widehat{F}) + D(G \,\|\, \widehat{G}) = o_p(n^{-b}) \tag{A.8}$$

by condition (iii). If $b = 1/2$, this implies $\sqrt{n}\,(m_n - \mu_0) \xrightarrow{p} 0$. Condition (ii) ensures $v_n \xrightarrow{p} \sigma_0^2$. Together with (A.7), we have

$$\frac{\sqrt{n}(\bar{R}-\mu_0)}{\sigma_0} \xrightarrow{D} N(0,1).$$

The result then follows by Slutsky's theorem as $s_R^2 \xrightarrow{p} v_n \xrightarrow{p} \sigma_0^2$, the first convergence due to condition (i).

If $b = 0$, inequality (A.8) implies that $m_n \xrightarrow{p} \mu_0$. Applying the weak law of large numbers to $\bar{R}$ conditioning on $(\widehat{F},\widehat{G})$, we have $\bar{R} - m_n \xrightarrow{p} 0$ and thus $\bar{R} \xrightarrow{p} \mu_0$.

□

A.2.3 Proof of Theorem 3

Proof. The intersection of the following three events imply that Algorithm 2 will recover the true DAG $\mathcal{G}_0$:

1. The algorithm recovers the true CPDAG of $\mathcal{G}_0$.
2. The independence measure ranks undirected edges that satisfy the PANM ahead of those that do not.
3. The likelihood ratio test returns the true orientation of an undirected edge that satisfying the PANM.

Note that sample splitting is used in the second and third events.

Together, the three events imply that the simplified Algorithm 2 coincides with its population version Algorithm 1, and thus, by Theorem 1, it outputs the true DAG $\mathcal{G}_0$. In the following three subsections, we prove each of the three events occurs with probability approaching one in the large sample limit, which completes the proof. □

A.2.3.1 Consistency of Initial CPDAG Learning

Our method uses the PC algorithm to learn the initial graph, which is achieved by iteratively finding a separating set S_{ij} to test the conditional independence between (X_i, X_j). Our implementation uses the partial correlation test. The consistency of this step is given by the following result.

Lemma 7 (Consistency of Initial Learning Algorithm)**.** *Suppose that the joint distribution of $(X_1, \ldots, X_p)$ is faithful to a DAG $\mathcal{G}_0$ and satisfies (A1) of Assumption 2. Let $\mathcal{E}$ be the CPDAG of $\mathcal{G}_0$ and let $\widehat{\mathcal{G}}$ be the graph estimated by the initial learning phase (Lines 1 –11 of Algorithm 2). For some choice of $\alpha_1, \alpha_2 \to 0, P(\widehat{\mathcal{G}} = \mathcal{E}) \to 1$ as $n \to \infty$.*

Proof. To prove this lemma, we show that the conditional independence tests are pointwise consistent. In the partial correlation test, let $\rho^*_{ij|S}$ and $\widehat{\rho}_{ij|S}$ respectively denote the true and estimated partial correlation values of $[X_i, X_j \mid S]$. Let

$$Z_n(i, j, S) = \frac{1}{2} \log \frac{1 + \widehat{\rho}_{ij|S}}{1 - \widehat{\rho}_{ij|S}}$$

be the Fisher z-transformation applied to the estimated partial correlation, and Z^* be defined similarly for $\rho^*_{ij|S}$. Under $H_0 : X_i \perp\!\!\!\perp X_j \mid S$, the quantity $\sqrt{n - |S| - 3}\, Z_n(i, j, S) \xrightarrow{D}$

$N(0,1)$ as $n \to \infty$. Let $Z_{1-\gamma_n/2}$ denote the critical value corresponding to the percentile $1-\gamma_n/2$ under the standard normal distribution. Under this asymptotic distribution, the probability of a type I error $P(|Z_n| > Z_{1-\gamma_n/2} \mid H_0) \to \gamma$ if the sequence $\gamma_n \to \gamma$ as $n \to \infty$.

By (A1) of Assumption 2, $|\rho^*_{ij|S}| > \tau > 0$ if $X_i \not\perp\!\!\!\perp X_j \mid S$. The power of the test is

$$\begin{aligned} & P\left(|Z_n(i,j,S)|\sqrt{n-|S|-3} > Z_{1-\gamma_n/2}\right) \\ >& P\left[\left(\frac{1}{2}\log\frac{1+\tau}{1-\tau} + O_p(n^{-1/2})\right)\sqrt{n-|S|-3} > Z_{1-\gamma_n/2}\right], \end{aligned}$$

since $|Z_n(i,j,S)| > \frac{1}{2}\log\frac{1+\tau}{1-\tau} + O_p(n^{-1/2})$. Observe that

$$\left(\frac{1}{2}\log\frac{1+\tau}{1-\tau} + O_p(n^{-1/2})\right)\sqrt{n-|S|-3} = C_\tau\sqrt{n} + O_p(1) \xrightarrow{p} \infty,$$

where $C_\tau > 0$ is some constant dependent on τ. Thus, we can choose $\gamma_n \to 0$ such that $Z_{1-\gamma_n/2} = o(\sqrt{n})$ and the power converges to 1 as $n \to \infty$.

We can specify the input significance levels as $\alpha_1 = \gamma_n$ and $\alpha_2 = (1+\delta)\gamma_n$ for some constant $\delta \geq 0$. By obtaining the correct skeleton and applying Meek's rules, the algorithm recovers the true CPDAG in the large-sample limit.

□

A.2.3.2 Consistency of Edge Ranking

In the edge orientation procedure, the algorithm ranks edges by the normalized, edge-wise mutual information in (2.3) to identify edges satisfying the PANM criterion. The calculation involves first discretizing variables $\varepsilon_{i,S}, X_k$, then computing the mutual information $\widehat{MI}(\varepsilon_{i,S}, X_k)$ for any $k \in S \subseteq \mathrm{pa}_{\mathcal{G}}(i) \cup \mathrm{ch}_{\mathcal{G}}(i)$ in PDAG $\mathcal{G}$. We first show the consistency of the mutual information estimator $\widehat{MI}$.

Suppose (X, Z) satisfy an additive regression model

$$X = m(Z) + \varepsilon,$$

where $Z = (Z_1, \ldots, Z_d)$ and ε could be dependent. Note that m and ε can be defined by (2.13) and (2.14). Fix an index $j \in \{1, \ldots, d\}$. Let $\{(x_i, z_i)\}_{i=1}^n$ be an i.i.d. test sample, and let

$$\widehat{\varepsilon}_i := x_i - \widehat{m}(z_i), \qquad i = 1, \ldots, n,$$

where $\widehat{m}$ is constructed from an independent training sample, so that $\widehat{m} \perp\!\!\!\perp \{(x_i, z_i)\}_{i=1}^n$.

Let $\{I_a\}_{a=1}^A$ be a fixed finite collection of bins partitioning $\mathbb{R}$ for ε, and let $\{J_b\}_{b=1}^B$ be a fixed finite collection of bins partitioning the support of Z_j. Define the (true) cell probabilities

$$p_{ab} := P(\varepsilon \in I_a,\ Z_j \in J_b), \qquad p_{a\cdot} := \sum_{b=1}^B p_{ab}, \qquad p_{\cdot b} := \sum_{a=1}^A p_{ab}.$$

Let $MI(\varepsilon, Z_j)$ denote the discretized mutual information computed from the true cell probabilities $\{p_{ab}\}$:

$$MI(\varepsilon, Z_j) = \sum_{a=1}^A \sum_{b=1}^B p_{ab} \log \frac{p_{ab}}{p_{a\cdot} p_{\cdot b}}.$$

Let $\widehat{p}_{ab}$ be the empirical frequency of $\{(\widehat{\varepsilon}_i, z_{ij})\}_{i=1}^n$ in the bin $I_a \times J_b$. Denote by $\widehat{MI}$ the plug-in mutual information estimator using the empirical frequencies $\{\widehat{p}_{ab}\}$.

We make two assumptions before presenting a result on the consistency of the mutual information estimator:

(C1) Let $\partial I := \cup_{a=1}^A \partial I_a$ be the set of ε-bin boundaries. Then

$$P(\varepsilon \in \partial I) = 0, \quad \text{and} \quad P\big(\text{dist}(\varepsilon, \partial I) \le t\big) \to 0 \text{ as } t \downarrow 0.$$

(C2) There exists $c_0 > 0$ such that $p_{ab} \geq c_0$ for all a, b.

Note that for Gaussian ε, (C1) holds for any finite binning.

Theorem 4 (Consistency of discretized MI based on test residuals)**.** *Assume (C1) and (C2) hold true. If $\widehat{m}$ is L^2 consistent in the sense that, for an independent draw $Z \perp\!\!\!\perp \widehat{m}$,*

$$\mathbb{E}\big[(\widehat{m}(Z) - m(Z))^2\big] \to 0, \tag{A.9}$$

then $\widehat{MI} \xrightarrow{p} MI(\varepsilon, Z_j)$ as $n \to \infty$.

Proof. Write $\Delta(z) := \widehat{m}(z) - m(z)$ and note that on the test sample

$$\widehat{\varepsilon}_i = x_i - \widehat{m}(z_i) = m(z_i) + \varepsilon_i - \widehat{m}(z_i) = \varepsilon_i - \Delta(z_i).$$

Fix a bin index a and define the indicator

$$U_{n,i}^{(a)} := \mathbf{1}\{\widehat{\varepsilon}_i \in I_a\}, \qquad U_i^{(a)} := \mathbf{1}\{\varepsilon_i \in I_a\}.$$

If $|\Delta(z_i)| \leq t$ and $\operatorname{dist}(\varepsilon_i, \partial I) > t$, then $\widehat{\varepsilon}_i = \varepsilon_i - \Delta(z_i)$ lies in the same ε-bin as ε_i, hence $U_{n,i}^{(a)} = U_i^{(a)}$. Therefore,

$$P\big(U_{n,i}^{(a)} \neq U_i^{(a)}\big) \leq P\big(|\Delta(Z)| > t\big) + P\big(\operatorname{dist}(\varepsilon, \partial I) \leq t\big),$$

where (ε, Z) is an independent draw (and $Z \perp\!\!\!\perp \widehat{m}$). Since $\mathbb{E}[\Delta(Z)^2] \to 0$, we have $\Delta(Z) \to 0$ in probability, so for any fixed $t > 0$, $P(|\Delta(Z)| > t) \to 0$. By (C1), $P(\operatorname{dist}(\varepsilon, \partial I) \leq t) \to 0$ as $t \downarrow 0$. Choosing $t = t_n \downarrow 0$ slowly, we obtain $P\big(U_{n,i}^{(a)} \neq U_i^{(a)}\big) \to 0$. Because there are finitely many bins, the same conclusion holds jointly over all $a = 1, \ldots, A$.

For each cell (a, b) define

$$V_{n,i}^{(ab)} := \mathbf{1}\{\widehat{\varepsilon}_i \in I_a,\ Z_{ij} \in J_b\}, \qquad V_i^{(ab)} := \mathbf{1}\{\varepsilon_i \in I_a,\ Z_{ij} \in J_b\}.$$

Then $|V_{n,i}^{(ab)} - V_i^{(ab)}| \le \mathbf{1}\{U_{n,i}^{(a)} \neq U_i^{(a)}\}$, so

$$\mathbb{E}\big|V_{n,i}^{(ab)} - V_i^{(ab)}\big| \le P(U_{n,i}^{(a)} \neq U_i^{(a)}) \to 0.$$

By Markov's inequality,

$$\left|\widehat{p}_{ab} - \frac{1}{n}\sum_{i=1}^{n} V_i^{(ab)}\right| = \left|\frac{1}{n}\sum_{i=1}^{n}(V_{n,i}^{(ab)} - V_i^{(ab)})\right| \xrightarrow{p} 0,$$

and hence for each fixed (a, b),

$$\widehat{p}_{ab} \xrightarrow{p} \frac{1}{n}\sum_{i=1}^{n} V_i^{(ab)} \xrightarrow{p} p_{ab}.$$

Because A, B are fixed and finite, we have joint convergence of the entire finite vector $(\widehat{p}_{ab})_{a,b}$ to $(p_{ab})_{a,b}$ in probability, and likewise for the marginals $\widehat{p}_{a\cdot} \to p_{a\cdot}$ and $\widehat{p}_{\cdot b} \to p_{\cdot b}$.

Under (C2), the mutual information function

$$\Phi\big((q_{ab})\big) = \sum_{a,b} q_{ab} \log \frac{q_{ab}}{q_{a\cdot} q_{\cdot b}}$$

is continuous on the compact set $\{q_{ab} \ge c_0,\ \sum_{a,b} q_{ab} = 1\}$. Therefore, by the continuous mapping theorem,

$$\widehat{MI} = \Phi\big((\widehat{p}_{ab})\big) \xrightarrow{p} \Phi\big((p_{ab})\big) = MI(\varepsilon, Z_j).$$

This concludes the proof. □

Theorem 4 shows that the estimated mutual information of the discretized variables (ε, Z_j) converges in probability to the true mutual information as $n \to \infty$. Assumption

(C1) holds for any absolutely continuous error distribution and any fixed finite discretization, since bin boundaries have Lebesgue measure zero. Since X_S and $\varepsilon_{i,S}$ in (2.14) follow absolute continuous distributions, (C1) holds. Assumption (C2) holds if there are no empty bins as stated in Assumption 2. Equation (2.17) in Assumption 1 implies (A.9). Thus, we have $\widehat{MI}(\varepsilon_{i,S}, X_k) \xrightarrow{p} MI(\varepsilon_{i,S}, X_k)$ for all $k \in S$ and all $i \in [p]$.

The ranking procedure sorts undirected edges in the PDAG $\mathcal{G}$ by an independence measure. We demonstrate that the ranking of undirected edges by Algorithm 3 ranks those satisfying the PANM before other edges in the large-sample limit.

Recall from Section 4.2 that the edge-wise independence measure $\widetilde{I}(X, Y)$ for random variables X, Y is calculated by Equation (2.2) (through discretization). Let $\widehat{\widetilde{I}}(X_i, X_j)$ be the estimator of $\widetilde{I}(X_i, X_j)$, which is calculated from the estimated mutual information $\widehat{MI}(\varepsilon_{v,S}, X_k)$ for $k \in S \subseteq \mathrm{pa}_{\mathcal{G}}(v) \cup \mathrm{ch}_{\mathcal{G}}(v)$ and $v \in \{i, j\}$ and the entropy measures $\widehat{H}(\cdot)$ of such terms. By Theorem 4, $\widehat{MI}(\widehat{\varepsilon}_{v,S}, X_k) \xrightarrow{p} MI(\varepsilon_{v,S}, X_k)$ for $k \in S$ and $v \in [p]$. The consistency of the entropy, $\widehat{H}(X) \xrightarrow{p} H(X)$, follows from a similar result, as $H(X) = MI(X, X)$ for a random variable X.

Therefore, if random variables (X_i, X_j) follow the PANM, then $\widehat{MI}(\cdot, \cdot) \xrightarrow{p} 0$ under the true causal direction and thus $\widehat{\widetilde{I}}(X_i, X_j) \xrightarrow{p} 0$ as $n \to \infty$. If (X_i, X_j) do not follow the PANM, then (A2) and (A3) of Assumption 2 imply that $\widehat{\widetilde{I}}(X_i, X_j) \xrightarrow{p} \widetilde{I}(X_i, X_j) > \delta/c_2 > 0$ as $n \to \infty$, where δ is the mutual information lower bound in (A2) and c_2 is the upper bound of entropy measures in (A3). This reasoning shows that our algorithm would rank all pairs of (X, Y) that satisfy the PANM ahead of those that do not satisfy.

A.2.3.3 Consistency of Likelihood Ratio Test

Suppose the true causal direction of an undirected edge $X - Y$ is $X \to Y$ and $[X, Y \mid Z_1, Z_2]$ satisfies the PANM criterion, where $Z_1 = PA_X$ and $Z_2 = PA_Y$. Denote the models for the two orientations by $F : X \to Y$ and $G : Y \to X$ and their estimators by $\widehat{F}, \widehat{G}$ which are independent of the test samples $\{(X_i, Y_i, Z_{1,i}, Z_{2,i})\}_{i=1}^n$. The LR test statistic $LR_n(\widehat{F}, \widehat{G})$ (2.9) can be written as

$$\frac{1}{n} LR_n(\widehat{F}, \widehat{G}) = \frac{1}{n} \sum_{i=1}^{n} \log \frac{\widehat{F}(X_i, Y_i \mid Z_{1,i}, Z_{2,i})}{\widehat{G}(X_i, Y_i \mid Z_{1,i}, Z_{2,i})} := \widehat{R}_X + \widehat{R}_Y,$$

a sum of two log-likelihood ratios with

$$\widehat{R}_X = \frac{1}{n} \sum_{i=1}^{n} \log \frac{\widehat{F}_X(X_i \mid Z_{1,i})}{\widehat{G}_X(X_i \mid Y_i, Z_{1,i})}, \quad \widehat{R}_Y = \frac{1}{n} \sum_{i=1}^{n} \log \frac{\widehat{F}_Y(Y_i \mid X_i, Z_{2,i})}{\widehat{G}_Y(Y_i \mid Z_{2,i})}.$$

Furthermore, let μ_X and μ_Y denote the population log-likelihood ratios. Note that F_X, F_Y, G_X, G_Y are all Gaussian regression models (A.1), where F_X and F_Y are true models. We first show that all three conditions of Theorem 2 are satisfied for both $\widehat{R}_X$ and $\widehat{R}_Y$ by verifying the assumptions of Lemmas 3, 4, and 5.

Verification of Condition (i) Assumption (i-1) in Appendix A.1.1 is implied by the finite $(4 + 2\eta)$-th moment requirement on the error variables $\varepsilon_{i,S}$ in Assumption 1. Assumption (i-2) is implied by (2.16) of Assumption 1, and (i-3) by (2.15). By Lemma 3, Condition (i) of Theorem 2 holds for both $\widehat{R}_X$ and $\widehat{R}_Y$.

Verification of Condition (ii) For both $(\widehat{R}_X, R_X)$ and $(\widehat{R}_Y, R_Y)$, assumptions (ii-1) to (ii-3) of Appendix A.1.2 are implied by the finite $(4 + 2\eta)$-th moments of the error variables, (2.17), and (2.18) in Assumption 1, respectively. By Lemma 4, Condition (ii) of Theorem 2 holds true for $\widehat{R}_X$ and $\widehat{R}_Y$.

Verification of Condition (iii) It is straightforward to see that (2.18) and (2.17) in Assumption 1 imply Assumptions (iii-1) and (iii-2) in Appendix A.1.3 for both $\widehat{R}_X$ and $\widehat{R}_Y$. Thus, by Lemma 5, Condition (iii) with $b = 0$ holds for both likelihood ratios.

Then, as an immediate consequence of Theorem 2,

$$\frac{1}{n} LR_n(\widehat{F}, \widehat{G}) = \widehat{R}_X + \widehat{R}_Y \xrightarrow{p} \mu_X + \mu_Y.$$

By Lemma 1, the causal direction $X \to Y$ is identifiable, which implies that $\mu_X + \mu_Y > 0$ and $\omega_*^2 > 0$. Therefore, together with $\widehat{\omega}_n^2 \xrightarrow{p} \omega_*^2$, we have

$$LR_n(\widehat{F}, \widehat{G})/(\sqrt{n}\widehat{\omega}_n) = a\sqrt{n} + o_p(\sqrt{n}) \xrightarrow{p} +\infty,$$

where $a > 0$ is a constant. This establishes the consistency of the likelihood ratio test for determining the true causal direction with the same α_1 as in the proof of Lemma 7.

A.3 Supplementary Numerical Results

A.3.1 Comparison with Nonlinear Constraint-based Algorithms

We compared SNOE with three constraint-based methods for nonlinear DAG learning, kPC [Gretton et al., 2009], NNCL [Wang and Zhou, 2021], and RESIT [Peters et al., 2014]. Note that, in general, kPC and NNCL produce a PDAG instead of a DAG. A significance level of $\alpha = 0.01$ is used for both algorithms as recommended in their paper or software tutorials. RESIT does not scale well to $p > 20$ nodes, as mentioned in their paper and in several other works that compare with this algorithm. Thus, we

did not include RESIT in this comparison since all networks in our experiment are of size $p \geq 24$.

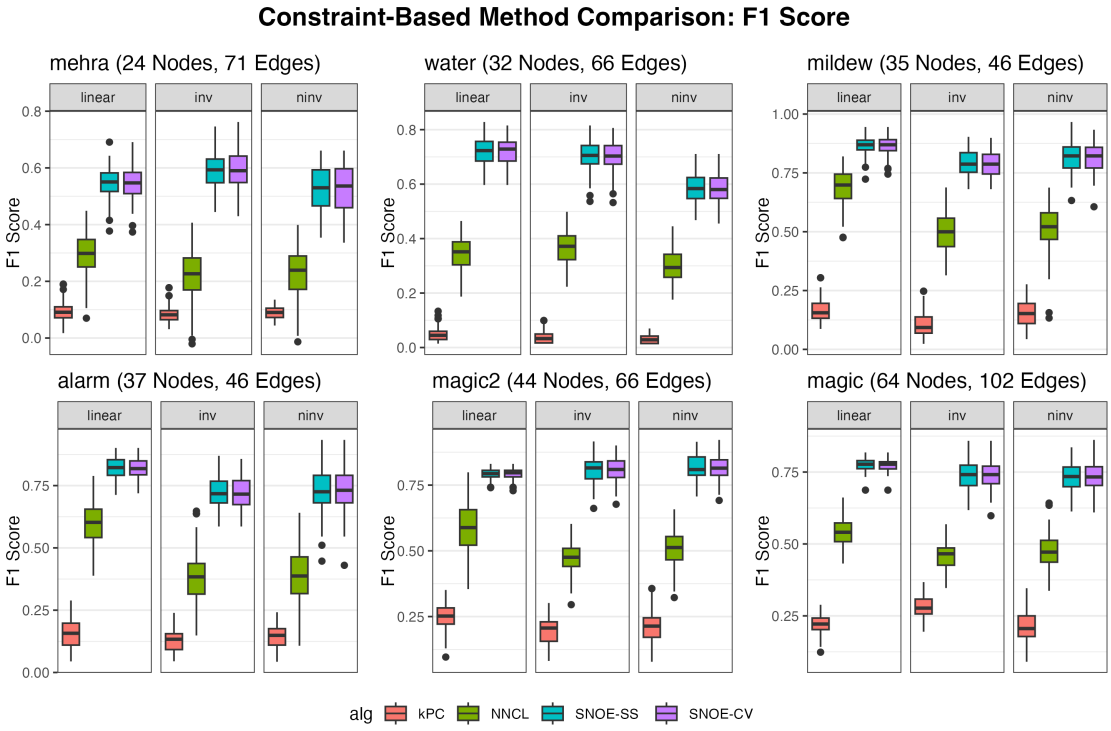


Figure A.3: F1 scores of various constraint-based methods.

Figure A.3 shows the F1 scores with respect to the true DAG. The results show that SNOE outperforms both kPC and NNCL under all settings. A detailed analysis reveals that kPC and NNCL have higher counts of extraneous and missing edges than SNOE. In particular, kPC has a greater amount of missing edges, likely due to sensitivity issues that stem from the HSIC test. The NNCL algorithm also uses the PC algorithm to estimate the CPDAG, yet its performance is most likely limited by the use of a piecewise linear function to approximate the underlying SEM. Both algorithms also contain a large number of incorrectly oriented and undirected edges

in the estimated PDAG. As SNOE outputs a PDAG without applying the final stage, we also analyzed the PDAG output from SNOE (not shown). We see that the PDAGs learned by SNOE still have a higher F1 score than the two competing methods. The F1 score is just slightly lower than that of the DAG output, which is also reflected in Figure 3.7 showing intermediate results. More detailed results show that there are still fewer undirected edges in the PDAG produced by SNOE than the other two methods. This reflects the advantageous design of our method to order edges satisfying the PANM criterion first, enabling correct identification of the causal direction with more oriented edges.

A.3.2 Performance Under General Nonlinear SEMs

We conducted additional simulations to assess the performance under more general functional forms. The data was simulated separately under invertible functions, non-invertible functions (Gaussian processes), and a multi-layer perceptron (MLP). The SEM under the invertible and non-invertible functions is

$$X_i = \sum_{j \in \mathrm{pa}(i)} f_j(X_j) + \sum_{k,m \in \mathrm{pa}(i)} X_k X_m + \varepsilon_i.$$

The function f_j is either an invertible or non-invertible function and up to five pairwise interaction terms are introduced in the SEM. For the MLP, the data generating function can be expressed as $X_i = W_2\sigma(W_1\mathrm{PA}_i) + \varepsilon_i$, where PA_i is regarded as a column vector corresponding to the k parent variables, $W_1 \in \mathbb{R}^{h\times k}$ and $W_2 \in \mathbb{R}^{1\times h}$ for $h = 200$, and $\sigma(\cdot)$ is the sigmoid function applied component-wise. Other settings remain the same as those detailed in Section 3.1.

In Figure A.4, we present a comparison of algorithm performances on data generated by general nonlinear functions. We observe an overall decrease in the F1

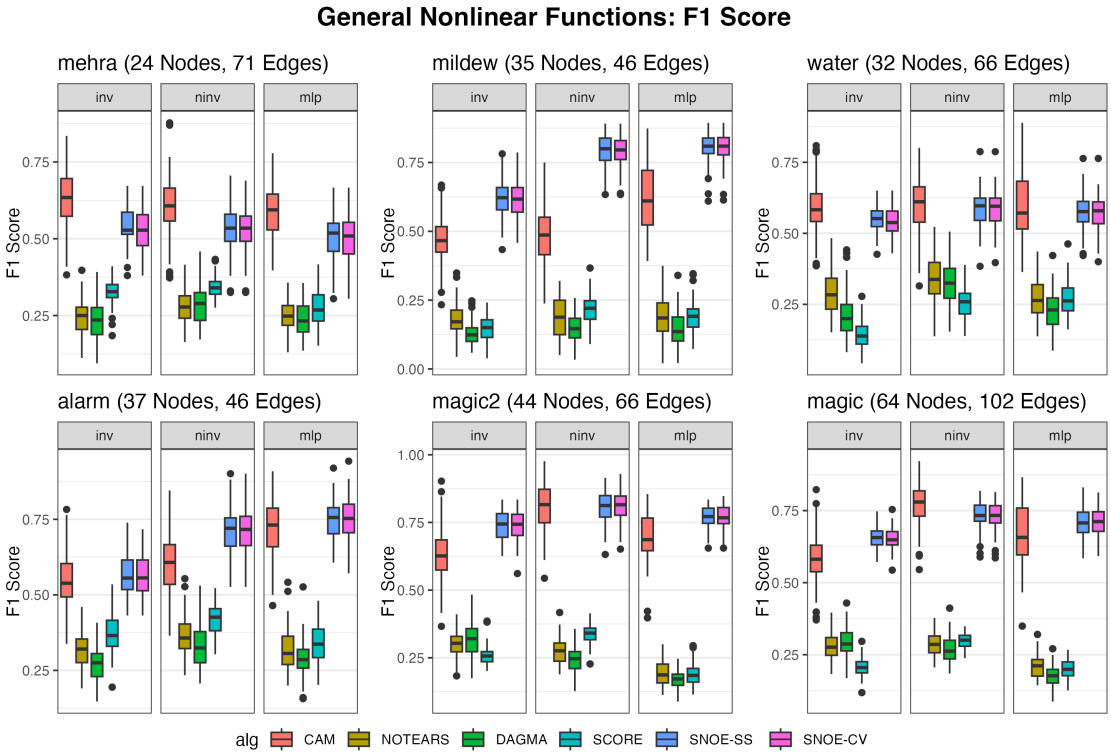


Figure A.4: Performance comparison under violations of the CAM assumption, where the data generating functions are not purely additive.

score compared to results on data generated under purely additive functions for all algorithms, as expected. A close analysis shows that SNOE has higher counts of false negatives, likely because the statistical power of the partial correlation test, utilized for learning the CPDAG, decreases when testing for more complex relations. While the difference in structural learning accuracies between SNOE and CAM is now smaller, SNOE still outperformed CAM in some cases. The F1 score for SNOE is more consistent and has a smaller interquartile range across all function types. CAM has fewer missing edges in comparison, but captures far more false positives. SNOE outperformed SCORE, NOTEARS, and DAGMA again in all cases.

SCORE, NOTEARS, and DAGMA have higher skeleton learning accuracies under these settings, as neural networks can better capture nonlinearities, but also learned many incorrectly oriented edges. Moreover, SCORE captured many false positives and NOTEARS and DAGMA contain many false negatives. Under these more generalized nonlinear functions, the small change in F1 scores and consistent performance across different functions indicate that SNOE is robust to violations of the functional form.

A.3.3 Accuracy of Initial Graph on Algorithm Performance

As SNOE improves upon a learned CPDAG, we investigate how the correctness of the initial graph affects the performance by testing our method separately on the exact, true CPDAG and the learned CPDAG. For the learned CPDAG, we split results by whether the F1 score, calculated with respect to the true CPDAG, is above the median across multiple data sets. The performances of the cross-validation (CV) and sample-splitting (SS) approaches are almost identical, so we only show that of the former. The results are shown in Figure A.5, with the F1 score of each approach calculated according to the true DAG for nonlinear functions and the true CPDAG for linear functions.

We first discuss the linear case when given the exact CPDAG, as one may notice that the F1 score decreases at the final stage. This is because we assume a nonlinear ANM and under this prior assumption, our algorithm extends the PDAG learned in step 3 to a DAG in the final step. Without this assumption, the algorithm would stop after stage 3, where the F1 score generally remains unchanged. Under the other nonlinear settings, we see indeed the initial CPDAG is imperative to learning the true DAG. Yet, the parallel F1 curves between the "AboveMedian" and "BelowMedian"

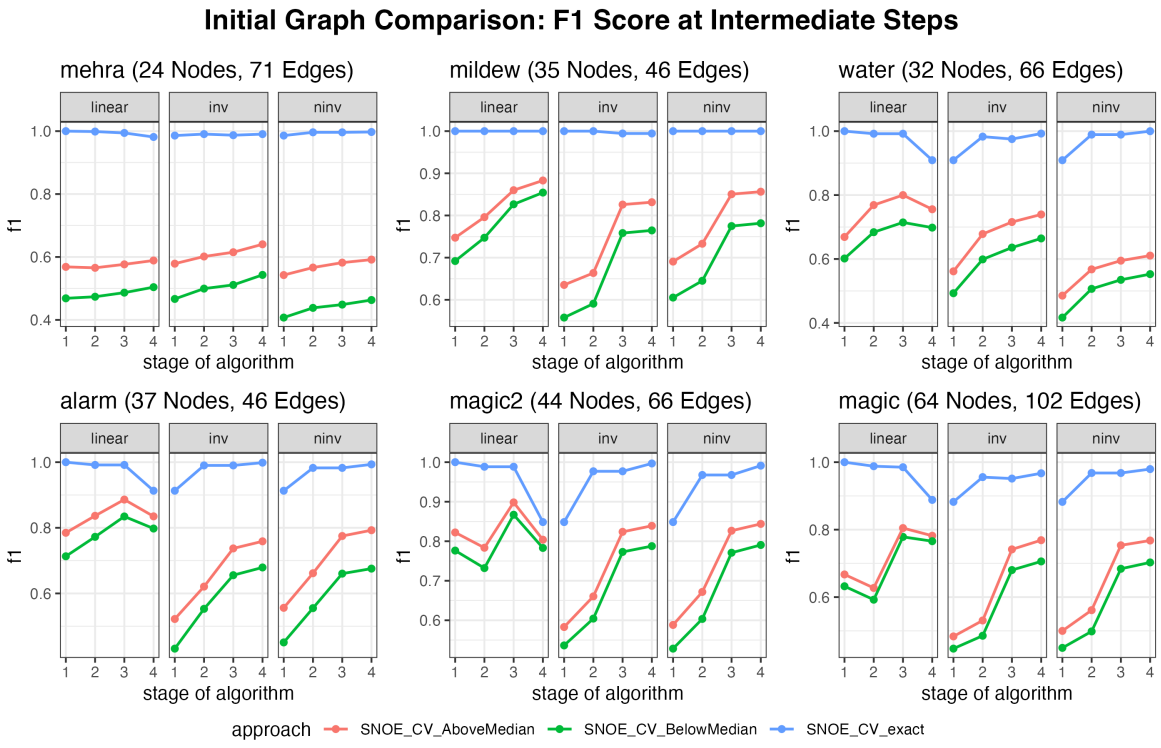


Figure A.5: F1 score of SNOE using different initial graphs: the true CPDAG ('exact'), a learned CPDAG with above median F1 score ('AboveMedian'), and a learned CPDAG with below median F1 score ('BelowMedian').

approaches indicate that an inaccurate estimation of a CPDAG does not exacerbate the difference in the final DAG learning accuracy. In fact, the results show that our subsequent steps are useful in learning the true DAG, whether using the true CPDAG or an estimated CPDAG as the starting point.

A.3.4 True Positives Captured at Intermediate Steps

The number of true positives captured at each stage of the SNOE algorithm is shown in Figure A.6 to complement and explain the intermediate F1 scores (see Figure 3.7). For data generated from nonlinear functions, the edge orientation procedure (step 2) uncovers a great number of true, directed edges from the learned CPDAG. Although the increase in the F1 score is relatively small at this stage since there are additional edges from the candidate set that may be false positives, the F1 score increases greatly once these edges are removed in the third stage. As for the linear, Gaussian DAG, the number of true positives remains unchanged, signaling high precision in the initial graph and correct inference on edge directions in subsequent stages. The results reflect the effectiveness of the edge orientation step in correctly orienting undirected edges. Moreover, the edge pruning step (step 3) deletes few if none true positives.

A.3.5 Runtime of Intermediate Steps

To breakdown the computational cost, we report the average runtime of each intermediate step in Figure A.7. Stages 1 (learning CPDAG by PC algorithm) and 4 (orienting graph into DAG per ANM assumption) exhibit similar runtimes across all networks. We utilized the PC-stable algorithm from the bnlearn package, which implements the iterative testing and orientation process in C++, and chose the partial correlation test for learning the skeleton, hence resulting in a lower runtime. The last stage orients remaining undirected edges in the PDAG, whose runtime is negligible relative to the other three stages. We observe that both stage 2 (edge orientation) and stage 3 (edge deletion) increase with network size, but at an approximately linear rate, indicating that our algorithm is scalable with graph size. SNOE-CV has a

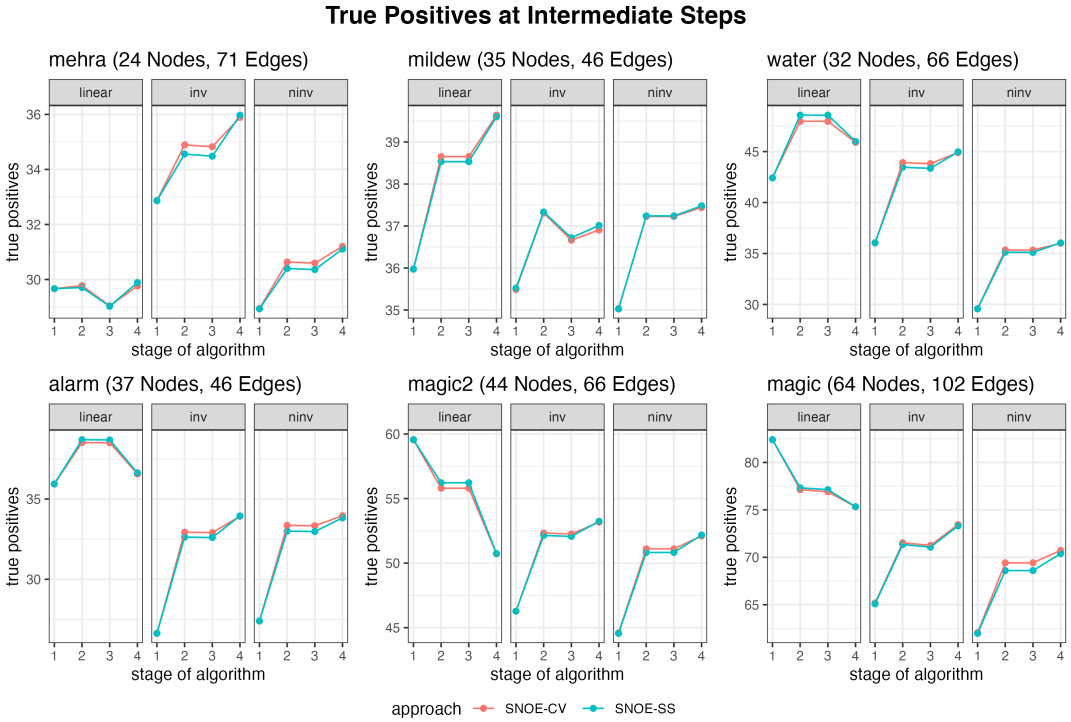


Figure A.6: Number of true positives captured. The edge orientation step (2) uncovers more edges and lifts the F1 score despite having extra candidate edges in the graph.

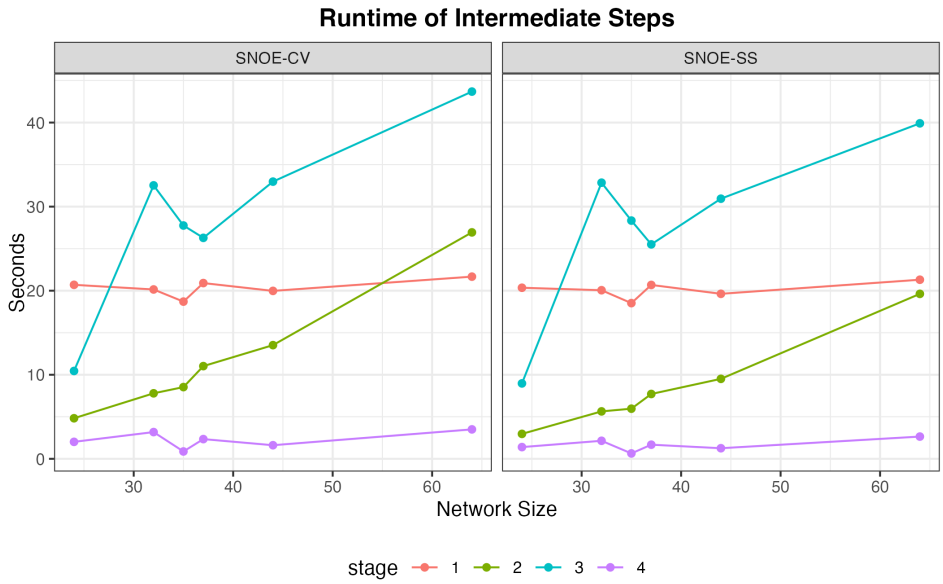


Figure A.7: Average runtime of intermediate steps by network size. The steps are (1) initial CPDAG learning, (2) edge orientation, (3) edge deletion, and (4) graph refinement.

slightly higher runtime in stage 2 since we employ cross validation in the likelihood test. Moreover, the use of Meek's rules assists in further detecting more directed edges, hence reducing the actual number of edges to evaluate.

A.3.6 Effect of Early Errors in Orientation Procedure

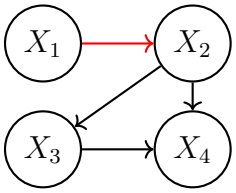
To study the downstream effect of an incorrect orientation by the likelihood ratio test in Algorithm 3, we compare the performance of the orientation procedure with early errors with that of when such errors are corrected. When applying the orientation procedure, the algorithm takes note of the first incorrect orientation made (on Line 7) during the procedure. Simultaneously in a separate process, the algorithm corrects

the edge direction and proceeds with the remaining steps (Line 8 and onward). There are no interventions thereafter. We then compare the learning accuracies of the graph before making the first incorrect orientation, the output graph of the procedure without intervention (original procedure), and the output graph of the procedure with the first error corrected. This comparison quantifies the negative effect of the first incorrect edge orientation. These graphs are labeled as *before_error*, *with_error*, and *fixed_error* in Table A.1. We used the same simulation specifications as Section 3.1.3 and tested the orientation procedure on $N = 300$ datasets per setting.

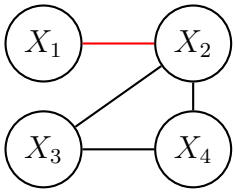
The downstream effect of the error is well-controlled, as evidenced by the small differences in the true positive (TP) and wrong orientation (Wrong Dir) counts in the table. If no downstream effects arose from the error, then the graph with the error fixed should have $\Delta\text{TP} = 1$ more true positive and $\Delta\text{Wrong Dir} = 1$ less incorrect orientation than the graph containing the error. We see that only one instance, network magic2 with invertible SEMs, has a difference of more than 2 true positives. The number of edges oriented by the procedure is slightly reduced in the presence of an error, as evidenced by a higher count of undirected edges (Undir). It is possible that some orientation rules on Lines 9 and 10 become inapplicable as a consequence of an earlier incorrect orientation. Nevertheless, the increase in undirected edges is less than 3 edges for the invertible setting and less than 2 edges under the non-invertible setting. Observing $\Delta\text{TP} \leq 2$ mostly and $\Delta\text{Wrong Dir} \leq 2$ for all cases, we can conclude that the downstream effect is minimal.

This reflects the key feature in our algorithm design to evaluate pairs of nodes sharing no neighbors first, as described on on Lines 2–4 of Algorithm 3, which minimizes the propagation of an error, should there be one. For instance, suppose the algorithm is applied to learn a DAG from its equivalence class, shown in Figure A.8

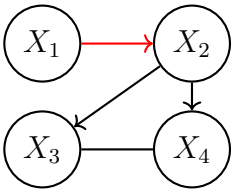
(a) and (b). The orientation procedure would first evaluate the edge $X_1 - X_2$, per the ranking procedure, and apply the likelihood ratio test. If the edge is oriented as $X_1 \to X_2$, then the procedure will apply Meek's rules to orient $X_2 \to X_3$ and $X_2 \to X_4$, depicted in (c). If the edge were incorrectly oriented as $X_2 \to X_1$ as in (d), neither Meek's rules nor the orientation on Line 9 of Algorithm 3 would apply since X_1 and X_2 share no neighbors. By this design, an error arising from an incorrect orientation does not easily propagate in the graph.



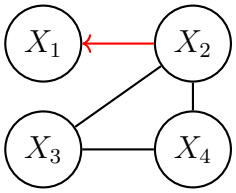
(a) DAG $\mathcal{G}$



(b) CPDAG of $\mathcal{G}$



(c) With correct orientation



(d) With incorrect orientation

Figure A.8: The design of Algorithm 3 can minimize further errors from an incorrect orientation. Under the correct orientation $X_1 \to X_2$, orientation rules are correctly applied and result in (c). On the converse shown in (d), the rules are not applicable and thus do not create any further errors.

A.3.7 Sensitivity to the Misranking of Edges

Given that the learned CPDAG is usually not perfect, it is difficult to supply a correct ranking of the undirected edges to the algorithm. Instead, we compare the results of orienting edges by our ranking as in SNOE-SS and SNOE-CV to those of orienting edges in an arbitrary order (SNOE-SS-Rand, SNOE-CV-Rand). We apply the approaches to data sets generated under linear, invertible(inv), and non-invertible(ninv) functions with Gaussian noise. The simulated data sets use the exact settings described in Section 6.3. Then, we examine F1 scores after the orientation stage of two select networks (mehra and magic2) in Figure 3.4 under the different functions. Table A.2 shows the F1 score after orienting edges in a random or ranked order, i.e. after stage 2 in Figure 3.7.

All approaches are applied to the same initial graph. The edge ranking procedure indeed led to more true positives in the orientation stage, as evidenced by the higher F1 scores. One may notice that the difference between the ranked and random procedures is not large. Under a random order, a greater proportion of undirected edges not satisfying the PANM criterion, due to missing parent variables, are evaluated by the likelihood test. As the algorithm cannot accurately estimate the true regression function due to missing parent variables, the likelihood test would likely find models for $X \to Y$ and $Y \to X$ equivalent, and thus keep the edge undirected. These edges will be correctly oriented later, once they meet the PANM condition following the orientation of other edges incident on the two nodes. In summary, the ranking procedure offers an improvement over a random ordering of edges. In the case where edges are misranked, the likelihood test prevents incorrect orientation and mitigates the impact of misranking.

Network	Approach	Data	Graph	F1	TP	Wrong Dir	Undir
mehra	LRT-CV	inv	before_error	0.446	26.528	6.696	4.857
			with_error	0.460	27.366	8.335	2.379
			fixed_error	0.483	28.745	7.932	1.404
mehra	LRT-CV	ninv	before_error	0.316	20.688	4.875	22.271
			with_error	0.388	25.458	11.573	10.802
			fixed_error	0.409	26.802	12.385	8.646
mehra	LRT-SS	inv	before_error	0.447	26.632	6.712	4.908
			with_error	0.462	27.491	8.362	2.399
			fixed_error	0.483	28.742	7.951	1.558
mehra	LRT-SS	ninv	before_error	0.312	20.405	4.216	23.252
			with_error	0.387	25.315	11.766	10.793
			fixed_error	0.407	26.631	12.108	9.135
magic2	LRT-CV	inv	before_error	0.510	42.612	2.225	10.969
			with_error	0.543	45.325	6.444	4.037
			fixed_error	0.587	49.025	5.631	1.150
magic2	LRT-CV	ninv	before_error	0.222	23.413	6.717	22.457
			with_error	0.250	26.435	15.370	10.783
			fixed_error	0.265	28.022	15.522	9.043
magic2	LRT-SS	inv	before_error	0.510	42.649	2.030	11.250
			with_error	0.542	45.375	6.494	4.060
			fixed_error	0.584	48.839	5.702	1.387
magic2	LRT-SS	ninv	before_error	0.226	23.829	7.158	20.974
			with_error	0.248	26.237	15.289	10.434
			fixed_error	0.260	27.487	15.079	9.395

Table A.1: F1 score and edge breakdown of the orientation procedure results: before making the first incorrect orientation (before_error), continued with the first incorrect orientation (with_error), and continued with the first incorrect orientation fixed (fixed_error).

network	approach	F1_linear	F1_inv	F1_ninv
mehra	SNOE-CV	0.516	0.551	0.504
mehra	SNOE-CV-Rand	0.514	0.536	0.504
mehra	SNOE-SS	0.524	0.554	0.510
mehra	SNOE-SS-Rand	0.512	0.533	0.499
magic2	SNOE-CV	0.761	0.631	0.639
magic2	SNOE-CV-Rand	0.744	0.628	0.620
magic2	SNOE-SS	0.766	0.632	0.631
magic2	SNOE-SS-Rand	0.753	0.618	0.612

Table A.2: F1 scores of the graphs after evaluating edges in a ranked or random manner in the orientation stage.

Bibliography

R Ayesha Ali, Thomas S Richardson, and Peter Spirtes. Markov equivalence for ancestral graphs. 2009.

Steen A Andersson, David Madigan, and Michael D Perlman. A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2): 505–541, 1997.

Matthew Ashman, Chao Ma, Agrin Hilmkil, Joel Jennings, and Cheng Zhang. Causal reasoning in the presence of latent confounders via neural admg learning. In *ICLR 2023*, May 2023.

Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. Dagma: Learning dags via m-matrices and a log-determinant acyclicity characterization. *Advances in Neural Information Processing Systems*, 35:8226–8239, 2022.

Carlos Brito and Judea Pearl. A new identification condition for recursive models with correlated errors. *Structural Equation Modeling*, 9(4):459–474, 2002.

Peter Bühlmann, Jonas Peters, and Jan Ernest. Cam: Causal additive models, high-dimensional order search and penalized regression. 2014.

Ruichu Cai, Jie Qiao, Kun Zhang, Zhenjie Zhang, and Zhifeng Hao. Causal discovery with cascade nonlinear additive noise model. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 1609–1615. International Joint Conferences on Artificial Intelligence Organization, 2019.

David Maxwell Chickering. Learning equivalence classes of bayesian-network structures. *The Journal of Machine Learning Research*, 2:445–498, 2002a.

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

Diego Colombo and Marloes H Maathuis. Order-independent constraint-based causal structure learning. *The Journal of Machine Learning Research*, 15(1):3741–3782, 2014.

Diego Colombo, Marloes H Maathuis, Markus Kalisch, and Thomas S Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pages 294–321, 2012.

Dorit Dor and Michael Tarsi. A simple algorithm to construct a consistent extension of a partially oriented graph. In *Technicial Report R-185, Cognitive Systems Laboratory, UCLA*, page 45. Citeseer, 1992.

Mathias Drton, Michael Eichler, and Thomas S Richardson. Computing maximum likelihood estimates in recursive linear models with correlated errors. *Journal of Machine Learning Research*, 10(10), 2009.

Robin J Evans. Graphs for margins of bayesian networks. *Scandinavian Journal of Statistics*, 43(3):625–648, 2016.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. *Advances in neural information processing systems*, 20, 2007.

Arthur Gretton, Peter Spirtes, and Robert Tillman. Nonlinear directed acyclic structure learning with weakly additive noise models. *Advances in neural information processing systems*, 22, 2009.

Uzma Hasan, Emam Hossain, and Md Osman Gani. A survey on causal discovery methods for iid and time series data. *arXiv preprint arXiv:2303.15027*, 2023.

Christina Heinze-Deml, Jonas Peters, and Nicolai Meinshausen. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2):20170016, 2018.

Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems*, 21, 2008.

Biwei Huang, Kun Zhang, Yizhu Lin, Bernhard Schölkopf, and Clark Glymour. Generalized score functions for causal discovery. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1551–1560, 2018.

Stella Huang and Qing Zhou. Nonlinear causal discovery through a sequential edge orientation approach. *arXiv preprint arXiv:2506.05590*, 2025.

Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H Maathuis, and Peter Bühlmann. Causal inference using graphical models with the r package pcalg. *Journal of statistical software*, 47:1–26, 2012.

Ilyes Khemakhem, Ricardo Monti, Robert Leech, and Aapo Hyvarinen. Causal autoregressive flows. In *International conference on artificial intelligence and statistics*, pages 3520–3528. PMLR, 2021.

Chun Li and Xiaodan Fan. On nonparametric conditional independence tests for continuous variables. *Wiley Interdisciplinary Reviews: Computational Statistics*, 12(3):e1489, 2020.

Zheng Li, Zeyu Liu, Feng Xie, Hao Zhang, Chunchen Liu, and Zhi Geng. Local identifying causal relations in the presence of latent variables. In *Forty-second International Conference on Machine Learning*, 2025.

Zhaolong Ling, Jiale Yu, Yiwen Zhang, Debo Cheng, Peng Zhou, Xingyu Wu, Bingbing Jiang, and Kui Yu. Local causal discovery without causal sufficiency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18737–18745, 2025.

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

Chris Meek. *Complete orientation rules for patterns*. Carnegie Mellon [Department of Philosophy], 1995.

Ricardo Pio Monti, Kun Zhang, and Aapo Hyvärinen. Causal discovery with general non-linear relationships using non-linear ica. In *Uncertainty in artificial intelligence*, pages 186–195. PMLR, 2020.

Joris M Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *Journal of Machine Learning Research*, 17(32):1–102, 2016.

Christopher Nowzohour, Marloes H Maathuis, Robin J Evans, and Peter Bühlmann.

Distributional equivalence and structure learning for bow-free acyclic path diagrams. 2017.

Juan Miguel Ogarrio, Peter Spirtes, and Joe Ramsey. A hybrid causal search algorithm for latent variable models. In *Conference on probabilistic graphical models*, pages 368–379. PMLR, 2016.

Judea Pearl. *Causality: models, reasoning, and inference*. Cambridge University Press, USA, 2000. ISBN 0521773628.

Jean-Philippe Pellet and André Elisseeff. Finding latent causes in causal networks: an efficient approach based on markov blankets. *Advances in Neural Information Processing Systems*, 21, 2008.

Jonas Peters, Joris M. Mooij, Dominik Janzing, and Bernhard Schölkopf. Identifiability of causal graphs using functional models. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, page 589–598, Arlington, Virginia, USA, 2011. AUAI Press. ISBN 9780974903972.

Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. 2014.

Joseph Ramsey, Bryan Andrews, and Peter Spirtes. Scalable causal discovery from recursive nonlinear data via truncated basis function scores and tests. *arXiv preprint arXiv:2510.04276*, 2025.

Alexander Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the simulated DAG! causal discovery benchmarks may be easy to game. In M. Ranzato,

A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 27772–27784. Curran Associates, Inc., 2021.

Thomas Richardson and Peter Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030, 2002.

Thomas S Richardson. A factorization criterion for acyclic directed mixed graphs. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 462–470, 2009.

Robert W Robinson. Counting unlabeled acyclic digraphs. In *Combinatorial Mathematics V: Proceedings of the Fifth Australian Conference, Held at the Royal Melbourne Institute of Technology, August 24–26, 1976*, pages 28–43. Springer, 2006.

Paul Rolland, Volkan Cevher, Matthäus Kleindessner, Chris Russell, Dominik Janzing, Bernhard Schölkopf, and Francesco Locatello. Score matching enables causal discovery of nonlinear additive noise models. In *International Conference on Machine Learning*, pages 18741–18753. PMLR, 2022.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations*, volume 9, 2021.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005. doi: 10.1126/science.1105809.

Marco Scutari. Learning bayesian networks with the bnlearn r package. *Journal of statistical software*, 35:1–22, 2010.

Rajen D Shah and Jonas Peters. The hardness of conditional independence testing and the generalised covariance measure. 2020.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

Ilya Shpitser, Robin J Evans, Thomas S Richardson, and James M Robins. Introduction to nested markov models. *Behaviormetrika*, 41(1):3–39, 2014.

Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1):62–72, 1991.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

Charles J. Stone. Additive regression and other nonparametric models. *Annals of Statistics*, 13:689–705, 1985.

Eric V Strobl, Kun Zhang, and Shyam Visweswaran. Approximate kernel-based conditional independence tests for fast non-parametric causal discovery. *Journal of Causal Inference*, 7(1):20180017, 2019.

Jin Tian and Judea Pearl. On the testable implications of causal models with hidden variables. 2002.

Kento Uemura and Shohei Shimizu. Estimation of post-nonlinear causal models using autoencoding structure. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3312–3316. IEEE, 2020.

Thomas Verma and Judea Pearl. Equivalence and synthesis of causal models. *Probabilistic and Causal Inference*, 1990.

Matthew J Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya like DAGs? a survey on structure learning and causal discovery. *ACM Computing Surveys*, 55(4): 1–36, 2022.

Quang H. Vuong. Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica*, 57(2):307–333, 1989. ISSN 00129682, 14680262.

Bingling Wang and Qing Zhou. Causal network learning with non-invertible functional relationships. *Computational Statistics & Data Analysis*, 156:107141, 2021.

Marcel Wienöbst, Max Bannach, and Maciej Liśkiewicz. Extendability of causal graphical models: Algorithms and computational complexity. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1248–1257. PMLR, 27–30 Jul 2021.

Simon N Wood. mgcv: Gams and generalized ridge regression for r. *R news*, 1(2): 20–25, 2001.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. In *International conference on machine learning*, pages 7154–7163. PMLR, 2019.

Jiji Zhang. Causal reasoning with ancestral graphs. *Journal of Machine Learning Research*, 9(7), 2008a.

Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17): 1873–1896, 2008b.

Jiji Zhang and Peter L Spirtes. A transformational characterization of markov equivalence for directed acyclic graphs with latent variables. *arXiv preprint arXiv:1207.1419*, 2012.

Kun Zhang and Aapo Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, page 647–655. AUAI Press, 2009. ISBN 9780974903958.

Kun Zhang and Aapo Hyvärinen. Nonlinear functional causal models for distinguishing cause from effect. *Statistics and causality: Methods for applied empirical research*, pages 185–201, 2016.

Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-based conditional independence test and application in causal discovery. *arXiv preprint arXiv:1202.3775*, 2012.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31, 2018.

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Learning

sparse nonparametric DAGs. In *International Conference on Artificial Intelligence and Statistics*, pages 3414–3425. Pmlr, 2020.

Qing Zhou. *Latent Structure and Causality: Inference from Data*. World Scientific, 2025.