LiNGAM Models on Minimal Latent Polytrees: A Cumulant–Based Discrepancy Approach

Sang Hyeon Lee
Supervisor: Prof. Dr. Mathias Drton
Mentor: Daniele Tramontano
Department of Mathematics, Technical University of Munich

February 22, 2026

Abstract

This thesis studies linear, non-Gaussian, acyclic models (LiNGAM) whose underlying directed graph is a *minimal latent polytree*. Building on moment identities from the LiNGAM literature, we propose a novel discrepancy matrix based on second and third order cumulants of the observed variables. We adapt the algorithm proposed by Etesami et al. to the LiNGAM setting, enabling provably consistent recovery of both observed and latent portions of the polytree. We outline theoretical guarantees and provide an empirical evaluation on synthetic data sets.

Contents

1 Introduction

Understanding causality—the fundamental question of which events or variables influence others—has been central to scientific inquiry across disciplines for centuries. From the experimental sciences that seek to establish cause-and-effect relationships through controlled interventions, to the social sciences that must often rely on observational data to infer causal mechanisms, the challenge of distinguishing genuine causal relationships from mere statistical associations remains one of the most important methodological problems in empirical research.

A modern framework for causal inference has its roots in the pioneering work of Pearl [2009], who formalized the distinction between statistical dependence and causal relationships through the lens of directed acyclic graphs (DAGs) and structural equation models. This framework recognizes that while correlation patterns can suggest possible causal structures, additional assumptions—whether experimental, distributional, or structural—are necessary to identify the direction and magnitude of causal effects from observational data alone.

In parallel, the field of graphical modeling has developed sophisticated methods for representing and learning dependency structures among random variables [Lauritzen, 1996, Edwards, 2000]. These approaches excel at capturing the conditional independence relationships that characterize the joint distribution of multivariate data, but they generally cannot distinguish between statistically equivalent models that encode different causal assumptions. This limitation, known as the Markov equivalence problem, represents a fundamental challenge for causal discovery from observational data.

Causal discovery refers to the algorithmic task of inferring causal structures—typically represented as directed graphs—directly from observational data, without access to experimental interventions. The goal is to recover the directed acyclic graph that encodes the true causal mechanisms generating the data, exploiting distributional or structural assumptions that render the causal model identifiable. Unlike traditional statistical modeling that focuses on prediction or association, causal discovery aims to uncover the asymmetric relationships that describe how changes in one variable would influence others under hypothetical manipulations.

Causal discovery from observational data emerges at the intersection of these two research traditions. The central challenge is to design methods that can recover not merely the statistical dependency structure, but the underlying causal mechanisms that generated the observed data. This requires exploiting additional constraints beyond those captured by conditional independence relationships—constraints that can break the symmetries inherent in purely associational models and reveal the directional nature of causal influences.

The *Linear Non-Gaussian Acyclic Model* (LiNGAM) [Shimizu et al., 2006] represents a breakthrough in this endeavor, demonstrating that non-Gaussianity of disturbances suffices to orient edges in a directed acyclic graph that is fully observed. By departing from the Gaussian assumption that underlies many classical approaches, LiNGAM methods can

achieve full causal identifiability from observational data alone, without requiring experimental interventions or temporal information. This result has profound implications for causal discovery in domains where controlled experiments are impractical or impossible, such as economics, where randomized interventions on national policy variables are infeasible, or epidemiology, where ethical constraints prohibit experimental manipulation of disease exposure.

However, many practical systems contain latent variables—unobserved confounders, mediators, or common causes—whose omission can severely distort causal inferences. The presence of latent variables introduces additional challenges beyond those encountered in fully observed settings, as these hidden factors can create spurious dependencies among observed variables and mask the true causal structure. Traditional approaches to latent variable modeling often require strong parametric assumptions or prior knowledge about the number and nature of unobserved factors.

This thesis investigates LiNGAMs whose causal structure forms a *minimal latent polytree*, following the terminology of Etesami et al. [2016]. Polytree structures—directed acyclic graphs whose underlying undirected graph is a tree—represent an important middle ground between the restrictive assumptions of fully observed models and the computational intractability of general latent variable models. While polytrees impose structural constraints that may not hold in all applications, they capture many realistic scenarios while remaining amenable to efficient algorithmic solutions.

The theoretical foundation for our approach builds on the axiomatic framework of Etesami et al. [2016], which develops a four-axiom *discrepancy matrix* framework that suffices to learn a latent polytree structure. However, their work instantiates the discrepancy matrix via directed-information estimators specifically designed for time series data rather than single-time-slice observational studies. In this work, we bridge LiNGAM identifiability results with the Etesami et al. framework by constructing a cumulant-based discrepancy measure that satisfies the required axioms while remaining applicable to cross-sectional data.

Related work. Our cumulant-based discrepancy approach is closely related to several recent developments in latent variable causal discovery. Tramontano et al. [2022] developed algorithms for learning linear non-Gaussian polytrees in the fully observed setting, providing the concentration inequalities for sample cumulants that underpin our finite-sample analysis. A complementary line of work uses *rank constraints of high-order cumulant matrices* rather than discrepancy-based methods for latent polytree learning [Cai et al., 2024]. This technique identifies latent structure by testing rank constraints on augmented cumulant matrices $\Psi^{(k)}_{Y;Z} = [C^{(k)}_{Y,Z} \mid C^{(k+1)}_{Y,Z}]$, which combine k-th and $(k+1)$-th order cumulants to enable rank testing even when the covariance matrix alone is rank-deficient. Their graphical implication results can distinguish between different structural patterns (atomic-fork, atomic-chain, atomic-Y) through rank conditions.

Contributions. The main contributions are:

(C1) Definition of the *Latent–LiNGAM Polytree Model* (Section 3), combining non–Gaussian noise with minimal latent structures.

(C2) Construction of a *cumulant discrepancy matrix* (Section 4) and proof that it satisfies the axioms of Etesami et al. [2016].

(C3) Adaptation of the Separation–Tree–Merger algorithm to the new discrepancy and consistency analysis (Section 5).

(C4) Empirical study evaluating orientation accuracy and sample complexity under varying latent proportions (Section 6).

Organisation. Section 2 reviews the requisite graph–theoretic and statistical preliminaries. The latent–LiNGAM model is formalised in Section 3. Section 4 introduces the cumulant discrepancy matrix and establishes its properties. Section 5 outlines a learning algorithm and sketches consistency proofs. Section 6 presents numerical experiments, and Section 7 concludes with directions for future work.

2 Linear Non-Gaussian Structural Causal Models

A directed graph (digraph) is a pair $G = (V, E)$, where V is the set of vertices and $E \subset V \times V$ is the set of directed edges. We let $V = [p] := \{1, \dots, p\}$. An element $(i, j) \in E$ may also be denoted by $i \to j$. A digraph G is acyclic (i.e., a DAG) if it does not contain any directed cycle: there is no sequence of vertices $i_0, \dots, i_k$ with $i_j \to i_{j+1} \in E$ for $j = 0, \dots, k-1$ and $i_0 = i_k$.

A path in G is a sequence of vertices $i_0, \dots, i_k$ such that $i_j \to i_{j+1} \in E$ or $i_{j+1} \to i_j \in E$ for all j. It is directed if all the arrows point in the same direction. A *polytree* is a DAG in which there is a unique path between any two vertices.

The *skeleton* of a DAG is the undirected graph obtained by replacing each directed edge by an undirected edge. Here, edges are denoted by $\{i, j\}$.

If $i \to j \in E$, then i is a parent of j, and j is a child of i. If G contains a directed path from i to j, then i is an ancestor of j and j is a descendant of i. The sets of parents, children, ancestors, and descendants of $i \in V$ are denoted by $\mathrm{pa}(i)$, $\mathrm{ch}(i)$, $\mathrm{an}(i)$, $\mathrm{de}(i)$, respectively. The *out-degree* of vertex i is the number of children of i, i.e., $\text{out-deg}(i) = |\mathrm{ch}(i)|$.

Definition 2.1 (Conditional independence)**.** *Two random variables X_i and X_j are* conditionally independent *given a set of variables X_C if*

$$P(X_i = x_i, X_j = x_j \mid X_C = x_C) = P(X_i = x_i \mid X_C = x_C) P(X_j = x_j \mid X_C = x_C)$$

for all values x_i, x_j, x_C such that $P(X_C = x_C) > 0$. We denote this relationship by $X_i \perp\!\!\!\perp X_j \mid X_C$.

More generally, for disjoint subsets $A, B, C \subset [p]$, the random vectors X_A and X_B are conditionally independent given X_C, written $A \perp\!\!\!\perp B \mid C$, if

$$P(X_A = x_A, X_B = x_B \mid X_C = x_C) = P(X_A = x_A \mid X_C = x_C) P(X_B = x_B \mid X_C = x_C)$$

for all values x_A, x_B, x_C such that $P(X_C = x_C) > 0$.

The joint distribution of X satisfies the *local Markov property* with respect to G if

$$\{i\} \perp\!\!\!\perp [p] \setminus (\mathrm{pa}(i) \cup \mathrm{de}(i)) \mid \mathrm{pa}(i) \quad \forall i \in [p].$$

The *Markov equivalence class* of G is the set of all DAGs that encode the same conditional independence relations, i.e., for which the set of distributions satisfying the local Markov property is the same. The conditional independence relationships encoded by a DAG can be characterized through the concept of *d-separation* [Pearl, 2009]: two sets of nodes A and B are d-separated by a set C in G if every path between A and B is blocked by C, where a path is blocked if it contains a node $v \in C$ that is not a collider on that path, or if it contains a collider w such that neither w nor any of its descendants are in C. Under the global Markov property, d-separation in the graph corresponds exactly to conditional independence in the distribution: $A \perp\!\!\!\perp B \mid C$ if and only if A and B are

d-separated by C in G. Two DAGs are Markov equivalent if and only if they have the same skeleton (underlying undirected graph) and the same set of v-structures (colliders). See Maathuis et al. [2019, Chap. 1] for further details.

Two fundamental challenges arise when learning causal structure from observational data. First, multiple DAGs can be statistically indistinguishable because they encode the same set of conditional independence constraints. For instance, the two DAGs with two nodes and one edge (so, $1 \to 2$ and $1 \leftarrow 2$) are in the same Markov equivalence class, and cannot be distinguished empirically without imposing further assumptions on the model. This motivates the need for additional identifying assumptions, such as the non-Gaussianity constraints exploited in LiNGAM models. To illustrate, consider a two-node graph: under Gaussianity, the models $X_2 = \lambda X_1 + \varepsilon_2$ (causal direction $1 \to 2$) and $X_1 = \tilde{\lambda} X_2 + \tilde{\varepsilon}_1$ (direction $2 \to 1$) can be statistically indistinguishable. However, with non-Gaussian noise, third-order cumulants break this symmetry. For the direction $1 \to 2$, we have

$$\frac{\Sigma_{1,2}}{\Sigma_{1,1}} = \frac{\mathcal{C}^{(3)}_{1,1,2}}{\mathcal{C}^{(3)}_{1,1,1}} = \lambda,$$

while for the reverse direction $2 \to 1$, we obtain

$$\frac{\Sigma_{1,2}}{\Sigma_{2,2}} = \frac{\mathcal{C}^{(3)}_{1,1,2}}{\mathcal{C}^{(3)}_{2,2,2}} = \tilde{\lambda}.$$

These equations exploit the asymmetry of the noise distributions—for symmetric distributions (in particular Gaussian noise), all third-order cumulants vanish ($\mathcal{C}^{(3)}_{i,i,i} = 0$), rendering both ratios undefined and destroying identifiability. Under non-Gaussian noise, by contrast, the diagonal entries $\mathcal{C}^{(3)}_{1,1,1}$ and $\mathcal{C}^{(3)}_{2,2,2}$ are both nonzero (they equal the third cumulants κ_1 and κ_2 of the respective noise variables), so both ratios are well-defined. Crucially, only one of the two equations can hold simultaneously: if the true direction is $1 \to 2$, then $\mathcal{C}^{(3)}_{1,1,1} = \kappa_1$ is the third cumulant of an *exogenous* source variable, whereas $\mathcal{C}^{(3)}_{2,2,2}$ mixes contributions from both ε_1 and ε_2 and does not equal κ_2 alone—so the second ratio fails. The direction for which the ratio of covariance to cumulant is consistent singles out the true causal order, enabling full identifiability from observational data alone [Shimizu et al., 2006].

Methods for causal discovery thus aim to either infer the Markov equivalence class or infer the DAG itself in a model class that renders the graph identifiable. The former approach focuses on recovering the *completed partially directed acyclic graph* (CPDAG), a mixed graph that encodes the causal information common to all members of a Markov equivalence class [Meek, 1995]. The latter scenario, which is the focus of this work, postulates that the considered models exhibit special properties that permit identification of the full graph [Shimizu et al., 2006].

2.1 Structural Equations

A structural equation model hypothesizes that every random variable in X is functionally related to its parent variables, i.e.,

$$X_i = f_i(X_{\mathrm{pa}(i)}, \varepsilon_i), \quad i \in V,$$

where the ε_i are independent noise terms and the f_i are measurable functions.

This framework provides a principled approach to modeling causation by encoding the fundamental insight that effects are generated by their causes [Peters et al., 2017]. The structural equations make explicit the *causal mechanism* by which each variable is generated: X_i is determined as a function of its direct causes $X_{\mathrm{pa}(i)}$ and an independent random disturbance ε_i. The mutual independence of the noise terms $\{\varepsilon_i\}_{i=1}^p$ reflects the causal sufficiency assumption: there are no unobserved common causes (confounders) that simultaneously influence multiple variables in the system.

The key conceptual advantage of this formulation is that it distinguishes between *seeing* and *doing* [Pearl, 2009]. The structural equations achieve this by representing causal mechanisms as autonomous modules: each equation $X_i = f_i(X_{\mathrm{pa}(i)}, \varepsilon_i)$ describes how nature generates X_i from its causes, independently of how other variables are generated. An intervention that sets X_i to a fixed value x_i is formally modeled by replacing the original equation for X_i with the constant assignment $X_i = x_i$, leaving all other structural equations unchanged. This surgery on the system yields a new *interventional distribution* $P(X \mid \mathrm{do}(X_i = x_i))$ that differs from the observational conditional distribution $P(X \mid X_i = x_i)$: the former captures the effect of forcing X_i to take value x_i and propagating this change through the causal mechanisms, while the latter merely conditions on observing $X_i = x_i$ without disrupting the generating process. Such interventional reasoning is essential for causal inference and policy evaluation.

If the f_i are linear, then we obtain a *linear structural equation model* (LSEM). An LSEM can be written in matrix form as

$$X = (I - \Lambda)^{-\top}\varepsilon, \tag{2.1}$$

where $\Lambda = (\lambda_{ij})$ with $\lambda_{ij} \neq 0$ only if $i \to j \in E$.

The linearity assumption, while restrictive, offers significant computational and theoretical advantages. Linear models admit closed-form solutions for interventional distributions and enable the use of powerful algebraic tools for structure learning. Moreover, in many applications, linear approximations provide reasonable first-order descriptions of complex causal relationships.

An LSEM constrains the dependence structure on the coordinates of X, but not the mean. Hence, when working with the LSEM, we may assume without loss of generality that $\mathbb{E}[\varepsilon_i] = 0$, which implies $\mathbb{E}[X_i] = 0$ for all $i \in V$.

Let $\varepsilon^{(2)} = (\mathbb{E}[\varepsilon_i \varepsilon_j])_{ij}$ be the covariance matrix of ε, which is a diagonal matrix by

independence, and write $\varepsilon_i^{(2)} := \mathbb{E}[\varepsilon_i^2] > 0$ for its ith diagonal entry. The covariance matrix of X is then the positive definite matrix

$$\Sigma = (I - \Lambda)^{-\top} \varepsilon^{(2)} (I - \Lambda)^{-1}. \tag{2.2}$$

Derivation of equation (2.2). From the structural equation (2.1), we have $X = (I-\Lambda)^{-\top}\varepsilon$. The covariance matrix of X is therefore

$$\begin{aligned}
\Sigma &= \mathbb{E}[XX^\top] \quad (\text{since } \mathbb{E}[X] = 0) && (2.3)\\
&= \mathbb{E}\left[(I - \Lambda)^{-\top} \varepsilon\varepsilon^\top (I - \Lambda)^{-1}\right] && (2.4)\\
&= (I - \Lambda)^{-\top} \mathbb{E}[\varepsilon\varepsilon^\top] (I - \Lambda)^{-1} && (2.5)\\
&= (I - \Lambda)^{-\top} \varepsilon^{(2)} (I - \Lambda)^{-1}, && (2.6)
\end{aligned}$$

where the third equality uses the linearity of expectation and the final equality follows from the definition of $\varepsilon^{(2)}$. □

This relationship between the structural parameters $(\Lambda, \varepsilon^{(2)})$ and the observed covariance matrix Σ is fundamental to the identifiability analysis that follows [see, e.g., Tramontano et al., 2022]. When the noise is Gaussian, equation (2.2) captures all distributional information, leading to the Markov equivalence problem discussed above. The non-Gaussian setting, which we explore next, breaks this equivalence and enables full structural recovery.

2.2 Cumulants in Gaussian and Non-Gaussian Models

Cumulants are alternative representations of moments of a distribution [Comon and Jutten, 2010]. Here, we formalize the definition in higher order settings and discuss their implications under Gaussian and non-Gaussian errors.

Definition 2.2 (Cumulant tensor)**.** *The kth cumulant tensor of a random vector $(X_1, \ldots, X_p)$ is the k-way tensor in $\mathbb{R}^{p\times\cdots\times p} \equiv (\mathbb{R}^p)^{\otimes k}$ whose entry in position $(i_1, \ldots, i_k)$ is the joint cumulant*

$$\operatorname{cum}(X_{i_1}, \ldots, X_{i_k}) := \sum_{(A_1,\ldots,A_L)} (-1)^{L-1}(L-1)!\,\mathbb{E}\left[\prod_{j\in A_1} X_j\right] \cdots \mathbb{E}\left[\prod_{j\in A_L} X_j\right],$$

where the sum is taken over all partitions $(A_1, \ldots, A_L)$ of the multiset $\{i_1, \ldots, i_k\}$.

In our context, the variables have mean 0, so

$$\begin{aligned}
\operatorname{cum}(X_i) &= \mathbb{E}[X_i] = 0, && (2.7)\\
\operatorname{cum}(X_{i_1}, X_{i_2}) &= \operatorname{Cov}[X_{i_1}, X_{i_2}] = \mathbb{E}[X_{i_1} X_{i_2}]. && (2.8)
\end{aligned}$$

More generally, the sum can be restricted to the partitions in which all blocks A_i have at least two elements. In particular,

$$\operatorname{cum}(X_{i_1}, X_{i_2}, X_{i_3}) = \mathbb{E}[X_{i_1} X_{i_2} X_{i_3}], \tag{2.9}$$

$$\operatorname{cum}(X_{i_1}, X_{i_2}, X_{i_3}, X_{i_4}) = \mathbb{E}[X_{i_1} X_{i_2} X_{i_3} X_{i_4}] - \mathbb{E}[X_{i_1} X_{i_2}]\mathbb{E}[X_{i_3} X_{i_4}] \tag{2.10}$$

$$- \mathbb{E}[X_{i_1} X_{i_3}]\mathbb{E}[X_{i_2} X_{i_4}] - \mathbb{E}[X_{i_1} X_{i_4}]\mathbb{E}[X_{i_2} X_{i_3}]. \tag{2.11}$$

The following powerful result dictates a simple condition that characterizes the Gaussianity of X.

Theorem 2.3 (Marcinkiewicz's theorem Marcinkiewicz, 1939)**.** *If there exists k such that $\operatorname{cum}(X_{i_1}, \ldots, X_{i_j}) = 0$ for all $j \geq k$, then $k = 3$ and X has a multivariate Gaussian distribution.*

This theorem establishes that non-Gaussian distributions necessarily have non-zero cumulants of order three or higher, making these higher-order moments essential for distinguishing between Gaussian and non-Gaussian models.

Lemma 2.4. *If the variables $\varepsilon_1, \ldots, \varepsilon_n$ are jointly independent, then $\operatorname{cum}(\varepsilon_{i_1}, \ldots, \varepsilon_{i_k}) = 0$ unless $i_1 = \cdots = i_k$.*

Definition 2.5 (Tucker product)**.** *Let $\mathcal{T}$ be a k-way tensor in $\mathbb{R}^{n_1 \times \cdots \times n_k}$ and let $A^{(j)} \in \mathbb{R}^{m_j \times n_j}$ be matrices for $j = 1, \ldots, k$. The* Tucker product *of $\mathcal{T}$ with the matrices $A^{(1)}, \ldots, A^{(k)}$ is the k-way tensor*

$$\mathcal{T} \bullet A^{(1)} \bullet \cdots \bullet A^{(k)} \in \mathbb{R}^{m_1 \times \cdots \times m_k}$$

with entries

$$\left(\mathcal{T} \bullet A^{(1)} \bullet \cdots \bullet A^{(k)}\right)_{i_1, \ldots, i_k} = \sum_{j_1=1}^{n_1} \cdots \sum_{j_k=1}^{n_k} \mathcal{T}_{j_1, \ldots, j_k} A^{(1)}_{i_1, j_1} \cdots A^{(k)}_{i_k, j_k}.$$

When all matrices are identical, $A^{(1)} = \cdots = A^{(k)} = A$, we write this as $\mathcal{T} \bullet [A]_{j=1}^{k}$.

Lemma 2.6. *Let the random vector X follow the LSEM from (2.1) with noise vector ε. Let $\mathcal{C}^{(k)}$ and $\varepsilon^{(k)}$ be the kth order cumulant tensors of X and ε, respectively. Then*

$$\begin{aligned} \mathcal{C}^{(k)} &= \varepsilon^{(k)} \bullet \left[(I - \Lambda)^{-1}\right]_{j=1}^{k} \\ &= \varepsilon^{(k)} \bullet (I - \Lambda)^{-1} \bullet \cdots \bullet (I - \Lambda)^{-1} \end{aligned}$$

is the Tucker product of $\varepsilon^{(k)}$ and k copies of $(I - \Lambda)^{-1}$.

Notice here that $\mathcal{C}^{(k)}$ reduces to (2.2) when $k = 2$.

See Comon and Jutten [2010] and references therein for proofs of Theorem 2.3 and Lemmas 2.4 and 2.6.

The next definition introduces the cumulant model obtained from the LSEM (2.1).

Definition 2.7. *Let $G = (V, E)$ be a DAG, and let $K \geq 2$ be an integer. The Kth cumulant model of G is the set of K-way tensors*

$$\mathcal{M}^{(K)}(G) = \{\varepsilon^{(K)} \bullet [(I - \Lambda)^{-1}]_{j=1}^{K} \ : \Lambda \in \mathbb{R}^E,\ \varepsilon^{(K)} \in (\mathbb{R}^p)^K \text{ diagonal}\}.$$

Here, $\mathbb{R}^E$ is the set of $p \times p$ matrices with support E. Further, the cumulants up to order K defined by G are modeled by

$$\mathcal{M}^{(\leq K)}(G) = \mathcal{M}^{(2)}(G) \times \cdots \times \mathcal{M}^{(K)}(G). \tag{2.12}$$

By Theorem 2.3, all multivariate Gaussian vectors X correspond to the zero element of $\mathcal{M}^{(K)}(G)$ for $k \geq 3$. We therefore restrict attention to non-Gaussian noise variables ε, where higher-order cumulants break Markov equivalence and enable full graph identifiability [Shimizu et al., 2006, 2011]. We will exploit this property algorithmically using the signal provided by higher cumulants; we do this by way of *treks*.

Definition 2.8 (Multi-Trek)**.** *A k-trek between vertices $i_1, \ldots, i_k \in V$ of a DAG $G = (V, E)$ is a collection of directed paths $T = (P_1, \ldots, P_k)$ in G that share the same source and have i_j as the sink of P_j for all j. The common source node is the top of the trek* $\mathrm{top}(T)$*. A trek is simple if the top node is the unique node on all the paths.*

We denote the set of k-treks between $i_1, \ldots, i_k$ by $\mathcal{T}(i_1, \ldots, i_k)$ and the set of simple treks by $\mathcal{S}(i_1, \ldots, i_k)$. See Figure 1 for an example.

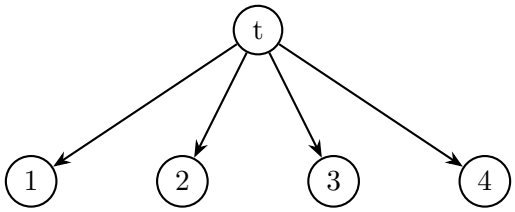


Figure 1: Example of a 4-trek.

If P is a directed path in the DAG $G = (V, E)$ and $\Lambda = (\lambda_{ij}) \in \mathbb{R}^E$, then $\lambda^P = \prod_{(i,j)\in P} \lambda_{ij}$ is a path monomial. For a k-trek $T = (P_1, \ldots, P_k)$, set $\lambda^T := \lambda^{P_1} \cdots \lambda^{P_k}$.

Proposition 2.9 (Multi-Trek Rule)**.** *The kth order cumulant tensor $\mathcal{C}^{(k)}(G)$ of X can be expressed as*

$$\mathcal{C}^{(k)}_{i_1,\ldots,i_k}(G) = \sum \varepsilon^{(k)}_{\mathrm{top}(T)} \lambda^T, \tag{2.13}$$

where the sum is over all the treks T in $\mathcal{T}(i_1, \ldots, i_k)$ and $\varepsilon^{(k)}_{\mathrm{top}(T)}$ denotes the $\mathrm{top}(T)$ *diagonal entry of $\varepsilon^{(k)}$.*

Proposition 2.9 follows from Lemma 2.6 and expanding the entries of $(I - \Lambda)^{-1}$ into sums of path monomials as in the usual trek rule for covariances [Robeva and Seby, 2021].

Corollary 2.10 (Simple Multi-Trek Rule)**.** *The kth order cumulant tensor $\mathcal{C}^{(k)}(G)$ of X can be expressed as*

$$\mathcal{C}^{(k)}_{i_1,\dots,i_k}(G) = \sum \mathcal{C}^{(k)}_{\mathrm{top}(S)}(G)\lambda^S, \tag{2.14}$$

where the sum is extended to all the simple treks S in $\mathcal{S}(i_1,\dots,i_k)$.

Corollary 2.11. *The ith diagonal entry of $\mathcal{C}^{(k)}$ is*

$$\mathcal{C}^{(k)}_i(G) = \sum_{p_1,\dots,p_k \in \mathrm{pa}(i)} \lambda_{p_1,i}\cdots\lambda_{p_k,i}\mathcal{C}^{(k)}_{p_1,\dots,p_k}(G) + \varepsilon^{(k)}_i.$$

2.3 Polytree Models

For general graphs, the algebraic relations among the cumulants may be far more complicated than the bivariate case (as illustrated in the two-node example of Tramontano et al. [2022, Example A.1]) and have not yet been fully characterized. However, there exists a generalization of rank-one constraints for polytrees, which we now discuss.

Since there is at most one directed path between any two nodes of a polytree G, there is at most one simple trek between any set of nodes $i_1,\dots,i_k$. The simple multi-trek rule then reduces to $C^{(k)}_{i_1,\dots,i_k}(G) = \lambda^S \mathcal{C}^{(k)}_{\mathrm{top}(S)}$ for a trek between nodes with S being the unique simple trek; denote the top of the simple trek between $i_1,\dots,i_k$, if it exists by $\mathrm{top}(i_1,\dots,i_k)$. Also, $C^{(k)}_{i_1,\dots,i_k}(G) = 0$ if there is no k-trek between the nodes.

For any two vertices $i \neq j$, let $c^{(i,j),k}_m$ denote the kth order cumulant $\mathcal{C}^{(k)}_{i\dots i,j\dots j}(G)$, where the first m indices are equal to i and the remaining $k-m$ equal j.

3 Latent–LiNGAM Polytree Model

We now specialize the general LiNGAM framework to the polytree setting with latent variables. This restriction provides significant computational advantages while still capturing important classes of causal relationships.

Definition 3.1 (Minimal latent polytree)**.** *Let $G = (V, E)$ be a DAG whose underlying undirected graph is a tree. Partition $V = O \cup L$ into observed and latent vertices. The pair (G, O) is a* minimal latent polytree *if every $\ell \in L$ has out-degree at least 2.*

The minimality condition ensures that no latent variable is redundant—removing any latent variable would disconnect the observed variables or change the conditional independence structure among them. This constraint is essential for identifiability, as latent nodes with out-degree 1 cannot be distinguished from direct edges between observed variables. To see why, consider a latent variable ℓ with parent X_j and single child X_i. The path $X_j \to \ell \to X_i$ induces the same conditional independence relationships among observed variables as the direct edge $X_j \to X_i$, rendering the latent node ℓ unidentifiable from observational data alone. Even in the LiNGAM setting with non-Gaussian noise, this

equivalence persists at the level of observed variables. Specifically, the model with latent variable

$$\begin{aligned} X_j &= \varepsilon_j, \\ \ell &= a_1 X_j + \varepsilon_\ell, \\ X_i &= a_2 \ell + \varepsilon_i \end{aligned}$$

is observationally equivalent to the direct edge model

$$\begin{aligned} X_j &= \varepsilon_j, \\ X_i &= a_2 a_1 X_j + (a_2 \varepsilon_\ell + \varepsilon_i), \end{aligned}$$

where the composite noise term $\tilde{\varepsilon}_i := a_2\varepsilon_\ell + \varepsilon_i$ is non-Gaussian whenever either ε_ℓ or ε_i is non-Gaussian. Thus, both models induce identical joint distributions over the observed variables (X_j, X_i), making it impossible to detect the presence of ℓ from observed data. This fundamental limitation motivates restricting attention to latent structures where each latent variable has at least two children, ensuring structural identifiability. This fundamental limitation motivates the minimality constraint in latent polytree models [cf. Etesami et al., 2016].

Definition 3.2 (Latent–LiNGAM polytree model)**.** *A random vector $X \in \mathbb{R}^{|V|}$ follows the latent–LiNGAM polytree model on (G, O) if:*

(i) The distribution of X satisfies the structural equation (2.1) with coefficient matrix Λ compatible with G.

(ii) The set $\varepsilon = (\varepsilon_i)_{i \in V}$ has independent, non–Gaussian entries with finite third moments.

(iii) Only $(X_i)_{i \in O}$ are observed.

This model combines the identifiability advantages of non-Gaussian noise with the computational tractability of polytree structures. The restriction to polytrees ensures that there is a unique undirected path between any two nodes, which simplifies both the theoretical analysis and algorithmic development.

3.1 Key Properties of the Model

Under the latent-LiNGAM polytree assumptions, several important properties hold:

(1) **Path uniqueness**: The unique directed path between any $i, j \in O$ factors through their lowest common ancestor (LCA). This property will underpin our discrepancy construction.

(2) **Moment identifiability**: The non-Gaussian noise assumption ensures that higher-order cumulants provide sufficient information to identify both the structure and parameters of the model, breaking the equivalence classes that arise under Gaussianity.

(3) **Computational tractability**: The polytree constraint reduces the complexity of structure learning algorithms from exponential (in general DAGs) to polynomial time.

The combination of these properties makes the latent-LiNGAM polytree model particularly well-suited for developing efficient structure learning algorithms based on cumulant information, as we will demonstrate in the subsequent sections.

4 Cumulant–Based Discrepancy Matrix

Definition 4.1 (Discrepancy on a polytree Etesami et al., 2016, Def. 7)**.** *Given a polytree $\vec{T} = (V, \vec{E})$ with root set R, every function $\gamma : V \times V \to \mathbb{R}$ that satisfies the following four criteria is called a* discrepancy *on $\vec{T}$:*

(1) $\gamma(v_1, v_2) = 0 \iff$ *either v_1 is an ancestor of v_2 or $v_1 = v_2$.*

(2) If $\mathrm{LCA}(v_1, v_2) = \mathrm{LCA}(v_1, v_3)$*, then* $\gamma(v_1, v_2) = \gamma(v_1, v_3)$*.*

(3) If $\mathrm{LCA}(v_1, v_2)$ *lies on the path from* $\mathrm{LCA}(v_1, v_3)$ *to v_1, then* $\gamma(v_1, v_2) < \gamma(v_1, v_3)$*.*

(4) $\gamma(v_1, v_2) < 0 \iff$ *v_1 and v_2 have no common ancestor.*

The image of such functions can be presented by the discrepancy matrix:

$$\Gamma_V := [\gamma(v_i, v_j)], \quad v_i, v_j \in V.$$

Note that for a given tree, the discrepancy matrix is not unique. Any function that satisfies the conditions in Definition 4.1 is a valid discrepancy measure.

Motivated by the rank conditions in Tramontano et al. [2022, Prop. 2.10], which show that certain cumulant matrices characterize edge orientations in fully observed polytrees, we define the following discrepancy measure between observed nodes.

Definition 4.2 (Cumulant Discrepancy Measure)**.** *For any pair of observed nodes $i, j \in O$, we define the cumulant discrepancy measure $\gamma(i, j)$ as*

$$\gamma(i,j) = \begin{cases} -1, & \text{if } \rho_{i,j} = 0, \\ 0, & \text{if } (\Sigma_{i,i}\mathcal{C}^{(3)}_{i,i,j} - \Sigma_{i,j}\mathcal{C}^{(3)}_{i,i,i}) = 0 \text{ or } i = j, \\ \dfrac{\mathcal{C}^{(3)}_{i,j,j}\,\Sigma_{i,i}}{\mathcal{C}^{(3)}_{i,i,j}\,\Sigma_{i,j}}, & \text{otherwise.} \end{cases} \tag{4.1}$$

where Σ is the covariance matrix, and $\mathcal{C}^{(3)}$ is the third-order cumulant tensor.

Remark 4.3. *Throughout this work, we restrict our analysis to third-order cumulants ($k = 3$). However, the extension to higher-order cumulants is straightforward: for any*

$k \geq 3$, *one can replace* $\mathcal{C}^{(3)}_{i,j,j}$ *with* $\mathcal{C}^{(k)}_{i,\underbrace{j,\ldots,j}_{k-1}}$ *and* $\mathcal{C}^{(3)}_{i,i,j}$ *with* $\mathcal{C}^{(k)}_{\underbrace{i,\ldots,i}_{k-1},j}$ *in* (4.1)*, following the same ratio structure. The choice of third-order moments provides a balance between identifiability power and estimation complexity in practice.*

Lemma 4.4 (Wright's Formula Wright, 1960)**.** *In a LiNGAM polytree model, the correlation* $\rho_{i,j} = Corr[X_i, X_j]$ *satisfies*

$$|\rho_{i,j}| = \begin{cases} \prod |\rho_e|, & \text{if } \mathcal{T}(i,j) \neq \emptyset, \\ 0, & \text{otherwise,} \end{cases} \tag{4.2}$$

where the product is taken over the edges e *of the unique trek connecting* i *and* j*, and* ρ_e *denotes the correlation between the variables at the endpoints of* e*.*

Remark 4.5. *This result implies that* $\rho_{i,j} = 0$ *if and only if there exists no trek connecting* i *and* j *in the polytree. This is particularly useful in interpreting the case* $\gamma(i,j) = -1$ *in Definition 4.2.*

Proposition 4.6 (Axioms)**.** *The map* $\gamma : O \times O \to \mathbb{R}$ *defined in Definition 4.2 satisfies the four axioms of Definition 4.1 on the latent–LiNGAM polytree model, provided the cumulants exist.*

Proof. We verify that the cumulant discrepancy measure $\gamma : O \times O \to \mathbb{R}$ defined in Definition 4.2 satisfies the four axioms of Definition 4.1, assuming all required cumulants exist.

(1) From Definition 4.2, $\gamma(i,i) = 0$. Assume $i \neq j$ and that i is an ancestor of j. Then we know that there is a simple trek between i and j. From the simple trek rule we know that $\Sigma_{i,j} = \lambda^S \Sigma_{i,i}$ and $\mathcal{C}^{(3)}_{i,i,j} = \lambda^S \mathcal{C}^{(3)}_{i}$, where S is the simple trek between i and j. Hence we have $(\Sigma_{i,i}\mathcal{C}^{(3)}_{i,i,j} - \Sigma_{i,j}\mathcal{C}^{(3)}_{i,i,i}) = 0$.

(2) Let v_i be the lowest common ancestor of two observed variables v_j and v_k. Suppose also that v_i is the lowest common ancestor of v_j and v_ℓ. We will show that

$$\gamma(j,k) = \gamma(j,\ell).$$

By Definition 4.2, for distinct observed vertices u, v one has

$$\gamma(u,v) = \frac{\mathcal{C}^{(3)}_{u,v,v}\,\Sigma_{u,u}}{\mathcal{C}^{(3)}_{u,u,v}\,\Sigma_{u,v}},$$

where Σ is the covariance matrix and $\mathcal{C}^{(3)}$ is the third-order cumulant tensor. We now express each entry in this ratio using the simple trek rule (Corollary 2.10).

(a) Because v_i is the lowest common ancestor of v_j and v_k, every simple trek from v_j to v_k factors through v_i. The simple trek rule implies

$$\mathcal{C}^{(3)}_{j,j,k} = \left(\lambda^{P(i,j)}\right)^2 \lambda^{P(i,k)}\, \mathcal{C}^{(3)}_i, \qquad \Sigma_{j,k} = \lambda^{P(i,j)}\, \lambda^{P(i,k)}\, \Sigma_i,$$

where $\lambda^{P(a,b)}$ denotes the product of structural coefficients along the unique directed path $P(a,b)$ in the polytree, and $\mathcal{C}^{(3)}_i$, Σ_i are the third-order cumulant and variance of X_{v_i}, respectively.

(b) Similarly,

$$\mathcal{C}^{(3)}_{j,k,k} = \lambda^{P(i,j)}\, \left(\lambda^{P(i,k)}\right)^2 \mathcal{C}^{(3)}_i, \qquad \Sigma_{j,j} = \left(\lambda^{P(i,j)}\right)^2 \Sigma_i.$$

Substituting these identities into the definition of γ gives

$$\begin{aligned}
\gamma(j,k) &= \frac{\mathcal{C}^{(3)}_{j,k,k}}{\mathcal{C}^{(3)}_{j,j,k}} \cdot \frac{\Sigma_{j,j}}{\Sigma_{j,k}} \\
&= \frac{\left[\lambda^{P(i,j)}\, (\lambda^{P(i,k)})^2\, \mathcal{C}^{(3)}_i\right] \cdot \left[\left(\lambda^{P(i,j)}\right)^2 \Sigma_i\right]}{\left[\left(\lambda^{P(i,j)}\right)^2 \lambda^{P(i,k)}\, \mathcal{C}^{(3)}_i\right] \cdot \left[\lambda^{P(i,j)}\, \lambda^{P(i,k)}\, \Sigma_i\right]} \\
&= 1.
\end{aligned}$$

An identical computation replacing v_k by v_ℓ yields

$$\gamma(j,\ell) = 1.$$

This completes the verification of Axiom (2).

(3) Suppose $d := \mathrm{LCA}(i,j)$ lies strictly below $c := \mathrm{LCA}(i,k)$ on the unique path from c to i. Write $\lambda^{P(u,v)}$ for the product of structural coefficients on the directed path $P(u,v)$ in the polytree. Using the simple–trek rule again, we obtain the following identities:

$$\begin{aligned}
\mathcal{C}^{(3)}_{i,j,j} &= \lambda^{P(d,i)}\, \left(\lambda^{P(d,j)}\right)^2 \mathcal{C}^{(3)}_d, \quad \mathcal{C}^{(3)}_{i,i,j} = \left(\lambda^{P(d,i)}\right)^2 \lambda^{P(d,j)}\, \mathcal{C}^{(3)}_d, \\
\mathcal{C}^{(3)}_{i,k,k} &= \lambda^{P(c,i)}\, \left(\lambda^{P(c,k)}\right)^2 \mathcal{C}^{(3)}_c, \quad \mathcal{C}^{(3)}_{i,i,k} = \left(\lambda^{P(c,i)}\right)^2 \lambda^{P(c,k)}\, \mathcal{C}^{(3)}_c, \\
\Sigma_{i,j} &= \lambda^{P(d,i)}\, \lambda^{P(d,j)}\, \Sigma_{d,d}, \qquad \Sigma_{i,k} = \lambda^{P(c,i)}\, \lambda^{P(c,k)}\, \Sigma_{c,c}.
\end{aligned}$$

Substituting into (4.1), the path monomials and third–order cumulants cancel, and

we find

$$\begin{aligned}\gamma(i,j) &= \frac{\mathcal{C}^{(3)}_{i,j,j}}{\mathcal{C}^{(3)}_{i,i,j}} \cdot \frac{\Sigma_{i,i}}{\Sigma_{i,j}} \\ &= \frac{\left[\lambda^{P(d,i)}\,(\lambda^{P(d,j)})^2\,\mathcal{C}^{(3)}_d\right] \cdot \Sigma_{i,i}}{\left[\left(\lambda^{P(d,i)}\right)^2 \lambda^{P(d,j)}\,\mathcal{C}^{(3)}_d\right] \cdot \left[\lambda^{P(d,i)}\,\lambda^{P(d,j)}\,\Sigma_{d,d}\right]} \\ &= \frac{\Sigma_{i,i}}{\left(\lambda^{P(d,i)}\right)^2 \Sigma_{d,d}},\end{aligned}$$

$$\begin{aligned}\gamma(i,k) &= \frac{\mathcal{C}^{(3)}_{i,k,k}}{\mathcal{C}^{(3)}_{i,i,k}} \cdot \frac{\Sigma_{i,i}}{\Sigma_{i,k}} \\ &= \frac{\left[\lambda^{P(c,i)}\,(\lambda^{P(c,k)})^2\,\mathcal{C}^{(3)}_c\right] \cdot \Sigma_{i,i}}{\left[\left(\lambda^{P(c,i)}\right)^2 \lambda^{P(c,k)}\,\mathcal{C}^{(3)}_c\right] \cdot \left[\lambda^{P(c,i)}\,\lambda^{P(c,k)}\,\Sigma_{c,c}\right]} \\ &= \frac{\Sigma_{i,i}}{\left(\lambda^{P(c,i)}\right)^2 \Sigma_{c,c}}.\end{aligned}$$

Taking the ratio gives

$$\frac{\gamma(i,j)}{\gamma(i,k)} = \frac{\left(\lambda^{P(c,i)}\right)^2 \Sigma_{c,c}}{\left(\lambda^{P(d,i)}\right)^2 \Sigma_{d,d}}.$$

Because c is an ancestor of d, the path from c to i factors through d. Hence $\lambda^{P(c,i)} = \lambda_{c,d}\,\lambda^{P(d,i)}$ and $\left(\lambda^{P(c,i)}\right)^2 = \lambda^2_{c,d}\left(\lambda^{P(d,i)}\right)^2$. Substituting yields

$$\frac{\gamma(i,j)}{\gamma(i,k)} = \frac{\lambda^2_{c,d}\,\Sigma_{c,c}}{\Sigma_{d,d}}.$$

Finally, using the Corollary 2.11 and that in a polytree there can be at most one simple trek between c and d, we have that $\Sigma_{d,d} = \lambda^2_{c,d}\,\Sigma_{c,c} + \omega^2_d$, where $\omega^2_d = \mathrm{var}(\varepsilon_d) > 0$ is the variance of the disturbance at d. Consequently,

$$\frac{\gamma(i,j)}{\gamma(i,k)} = \frac{\lambda^2_{c,d}\,\Sigma_{c,c}}{\lambda^2_{c,d}\,\Sigma_{c,c} + \omega^2_d} < 1,$$

proving that $\gamma(i,j) < \gamma(i,k)$ whenever $\mathrm{LCA}(i,j)$ is strictly below $\mathrm{LCA}(i,k)$.

(4) If i and j have no common ancestor, then there exists no trek between them. Hence from Lemma 4.4 we have $\rho_{i,j} = 0$, and thus $\gamma(i,j) = -1 < 0$.

□

4.1 Example: Discrepancy Matrix on a Four-Node Polytree

We illustrate the cumulant-based discrepancy measure on a simple polytree with four observed nodes. Let $V = \{v_1, v_2, v_3, v_4\}$ and consider the directed edges

$$v_1 \longrightarrow v_2, \qquad v_1 \longrightarrow v_3, \qquad v_3 \longrightarrow v_4,$$

depicted in Figure 2. In this example every vertex is observed; there are no latent variables.

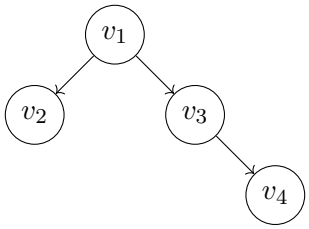


Figure 2: A four-node polytree with edges $v_1 \to v_2$, $v_1 \to v_3$ and $v_3 \to v_4$.

Structural parameters. Assign structural coefficients $\lambda_{1,2} = 2$, $\lambda_{1,3} = 3$ and $\lambda_{3,4} = 4$. Let the noise variables ε_i be independent with variances $\sigma_i^2 = 1$ and third cumulants $\kappa_i = 1$ for $i = 1, 2, 3, 4$. The structural equations are then

$$X_1 = \varepsilon_1, \quad X_2 = 2\,X_1 + \varepsilon_2, \quad X_3 = 3\,X_1 + \varepsilon_3, \quad X_4 = 4\,X_3 + \varepsilon_4 = 12\,X_1 + 4\,\varepsilon_3 + \varepsilon_4.$$

Covariances and third cumulants. From these recursions one computes the covariance matrix

$$\Sigma = \begin{pmatrix} 1 & 2 & 3 & 12 \\ 2 & 5 & 6 & 24 \\ 3 & 6 & 10 & 40 \\ 12 & 24 & 40 & 161 \end{pmatrix}.$$

The third-order cumulant tensor $\mathcal{C}^{(3)}$ has diagonal entries

$$\mathcal{C}^{(3)}_{1,1,1} = 1, \quad \mathcal{C}^{(3)}_{2,2,2} = 2^3 + 1 = 9, \quad \mathcal{C}^{(3)}_{3,3,3} = 3^3 + 1 = 28, \quad \mathcal{C}^{(3)}_{4,4,4} = (12)^3 + (4)^3 + 1 = 1793,$$

using the simple trek rule, where for $\mathcal{C}^{(3)}_{4,4,4}$, it can also be shown differently with Corollary 2.11 :

$$\mathcal{C}^{(3)}_{4,4,4} = \lambda^3_{3,4} \cdot \mathcal{C}^{(3)}_{3,3,3} + \kappa_4 = 4^3 \cdot 28 + 1 = 1793.$$

Moreover, with a polytree, the only non–vanishing mixed entries are of the form

$$\mathcal{C}^{(3)}_{i,j,k} \;=\; \lambda^{P(h,i)} \lambda^{P(h,j)} \lambda^{P(h,k)} \kappa_h,$$

where $\lambda^{P(h,i)}$ denotes the path monomial from h to i, with h being the unique common ancestor of i,j and k. For instance, $\mathcal{C}^{(3)}_{2,3,3} = 2 \cdot 3^2 \cdot 1 = 18$ and $\mathcal{C}^{(3)}_{2,3,4} = 2 \cdot 3 \cdot 12 \cdot 1 = 72$.

Discrepancy matrix. Applying Definition 4.2 yields the following cumulant discrepancy matrix $\Gamma = [\gamma(v_i, v_j)]$:

$$\Gamma = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{5}{4} & 0 & \frac{5}{4} & \frac{5}{4} \\ \frac{10}{9} & \frac{10}{9} & 0 & 0 \\ \frac{161}{144} & \frac{161}{144} & \frac{161}{160} & 0 \end{pmatrix}.$$

Here, each non-zero entry was computed using the ratio in (4.1) or, equivalently, by using the simplified formula derived in the proof of Proposition 4.6. For example,

$$\gamma(v_2, v_3) = \frac{\Sigma_{2,2}\, \mathcal{C}^{(3)}_{2,3,3}}{\mathcal{C}^{(3)}_{2,2,3}\, \Sigma_{2,3}} = \frac{5 \cdot 18}{12 \cdot 6} = \frac{5}{4}, \quad \gamma(v_4, v_3) = \frac{\Sigma_{4,4}}{\lambda^2_{3,4}\, \Sigma_{3,3}} = \frac{161}{4^2 \cdot 10} = \frac{161}{160} \approx 1.006.$$

Interpretation. The matrix Γ respects all four axioms of Definition 4.1. Zero entries arise whenever the first argument is an ancestor of the second; equal values appear when pairs share the same lowest common ancestor, and nested ancestry leads to increasing values, e.g. $\gamma(v_4, v_3) = 161/160 < \gamma(v_4, v_2) = 161/144$ since $\mathrm{LCA}(v_4, v_3) = v_3$ lies below $\mathrm{LCA}(v_4, v_2) = v_1$.

5 Recovery of Latent Trees

5.1 Theoretical Foundation

Before presenting the algorithmic framework, we establish the key theoretical concepts from Etesami et al. [2016] that underpin our approach.

Definition 5.1 (Learnable subset Etesami et al., 2016, Def. 8)**.** *In a polytree $\vec{T} = (V, \vec{E})$, we call a subset $L \subset V$* learnable *if every node $v \in L$ has at least two outgoing arrows. We call $O := V \setminus L$ the set of observed nodes.*

Remark 5.2. *From Definition 5.1, if L is a learnable subset of a polytree, then all the leaves belong to $O = V \setminus L$. This ensures that latent variables with insufficient connectivity (out-degree less than 2) cannot be distinguished from observed variables based solely on the discrepancy patterns.*

The following theorem provides the theoretical guarantee for the recovery algorithm:

Theorem 5.3 (Structure identifiability Etesami et al., 2016, Thm. 4)**.** *Let $\vec{T} = (V, \vec{E})$ be a polytree with root set R, and let $L \subseteq V$ be a learnable subset. Then the existence of a discrepancy matrix Γ_O for $O = V \setminus L$ suffices for learning $\vec{T}$.*

Proof. See Etesami et al. [2016, Appendix H]. □

The proof of Theorem 5.3 proceeds by mathematical induction on the size of the observed node set O and provides the constructive framework for the three-phase algorithm presented below.

Definition 5.4 (Tree merger Etesami et al., 2016, Def. 9)**.** *A* tree merger *is an operator that takes two directed trees $\vec{T}_1$, $\vec{T}_2$ and a given subtree of both of them, say $\vec{T}_3$, and merges them at $\vec{T}_3$. We denote this operation by*

$$\vec{T}_1 \circ \vec{T}_2|_{\vec{T}_3}.$$

Figure 3 demontrates one such tree merger.

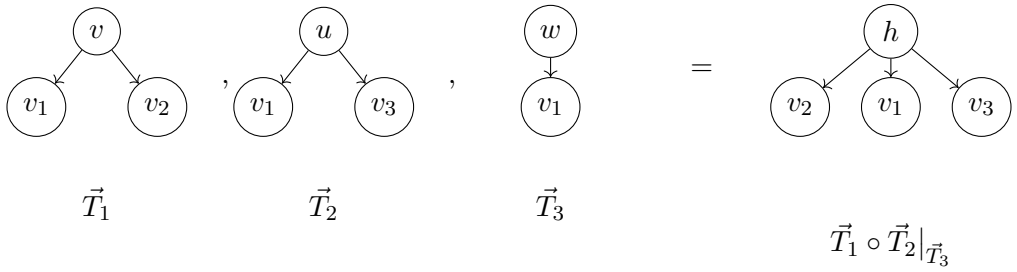


Figure 3: Illustration of the tree merger operation. Two directed trees $\vec{T}_1$ and $\vec{T}_2$ sharing a common subtree $\vec{T}_3$ (node v_1 with its parent) are merged to form a single polytree. The merger introduces a latent node h that becomes the common root with direct connections to all observed nodes.

5.2 Structure Recovery Algorithm

The rationale of our algorithmic approach follows the three main steps of the proof of Theorem 5.3:

(1) **Root discovery**: Discover the number of roots $|R|$ of the underlying polytree and all their descendants in the set of observed nodes O given the discrepancy matrix Γ_O. This can be accomplished by fixing a node $v \in O$ and finding a maximal subset of O containing v in which every pair of nodes has non-negative discrepancy.

(2) **Subtree recovery**: Recover the underlying tree for each root $r \in R$, based on the descendants of r that were discovered in O during the first step.

(3) **Tree merging**: Merge the trees recovered in the previous step to reconstruct the underlying polytree. When two recovered trees are connected, their combined subgraph forms a tree, which can be learned using the Tree algorithm.

The correctness of this approach relies on the key insight that if a polytree $\vec{T}$ and a directed tree $\vec{T}_i$ have a non-empty intersection, their union is guaranteed to be a single-rooted tree, enabling recursive reconstruction via the tree merger operation.

5.3 Algorithmic Recovery from Cumulant Discrepancy

We adapt the recovery pipeline of Etesami et al. [2016] to our cumulant-based discrepancy measure. The three-stage method first partitions nodes into sibling groups, then orients edges within groups, and finally inserts latent nodes to recover a minimal latent polytree.

Algorithm 1 Separation(Γ_O)

Require: Discrepancy matrix Γ_O
Ensure: Sibling groups $O_1, \dots, O_{|\mathcal{R}|}$

$M \leftarrow \varnothing$, $i \leftarrow 1$
while $O \setminus M \neq \varnothing$ **do**
  Choose $v \in O \setminus M$
  Find all $C \subseteq O$ such that $v \in C$ and
  for all $(u, w) \in C \times C$, $\gamma(u, w) \geq 0$
  $O_i \leftarrow$ maximal such C
  return O_i
  $M \leftarrow M \cup O_i$
  $i \leftarrow i + 1$

Algorithm 2 Tree(Γ_O)

Require: Discrepancy matrix Γ_O
Ensure: Directed tree $\vec{T} = (V, \vec{E})$

for all $v \in O$ **do**
  $B_v \leftarrow \arg\min_{u \in O \setminus \{v\}} \gamma(v, u)$
if $B_v = O \setminus \{v\}$ for all $v \in O$ **then**
  if $\exists w \in O$ such that $\min_{u \in O \setminus \{w\}} \gamma(w, u) = 0$ **then**
    $\vec{T}$ is a star graph with w as the root in the center
  else
    $\vec{T}$ is a star graph with a hidden node as the root in the center
else
  Choose w such that $B_w \neq O \setminus \{w\}$
  $\vec{T}' \leftarrow \textsc{Tree}(B_w \cup \{w\})$
  $\vec{T}'' \leftarrow \textsc{Tree}(O \setminus B_w)$
  Substitute w in $\vec{T}''$ by another node, say h
  $\vec{T} \leftarrow \vec{T}' \oplus \vec{T}''(h)$

Algorithm 3 Polytree(Γ_O)

Require: Discrepancy matrix Γ_O

Ensure: Minimal latent polytree $\vec{T} = (V, \vec{E})$

$\{O_1, \ldots, O_{|\mathcal{R}|}\} \leftarrow$ Separation(Γ_O)
$\vec{T} \leftarrow$ Tree(O_1)
$S \leftarrow O_1, \quad I \leftarrow \{1\}$
while $I \neq \{1, 2, \ldots, |\mathcal{R}|\}$ **do**
    Find $i \in \{1, \ldots, |\mathcal{R}|\} \setminus I$ such that $O_i \cap S \neq \varnothing$
    $\vec{T}_{\text{sub}} \leftarrow$ Tree($S \cap O_i$)
    $\vec{T}_i \leftarrow$ Tree(O_i)
    $\vec{T} \leftarrow \vec{T} \circ \vec{T}_i|_{\vec{T}_{\text{sub}}}$
    $S \leftarrow S \cup O_i$
    $I \leftarrow I \cup \{i\}$

Algorithm 3 presents the main algorithm for learning the polytree $\vec{T}(V, \vec{E})$ with the root set R given the discrepancy matrix Γ_O on its observed nodes O. First, it calls the subroutine Separation(Γ_O), which finds subsets O_is, where $O = \cup_i O_i$ such that each subset corresponds to observed nodes in a directed tree with a single root. Each of these single rooted subtrees can be learned by Algorithm 2. To complete the task, Algorithm 3 must connect these subtrees to recover the original polytree. This is done by using the fact that if a polytree $\vec{T}$ and a directed tree $\vec{T}_i$ have an intersection, their common subgraph is also a tree; thus, it can be learned using Algorithm 2.

Algorithm Description. The Separation algorithm operates on a given discrepancy matrix Γ_O of the observed nodes. The aim of this partition is to obtain the set of vertices O into subsets $O_1, O_2, \ldots, O_{|R|}$, where $|R|$ is the total number of subsets corresponding to different roots. Each subset satisfies a specific compatibility condition $\gamma(u, w) \geq 0$ for all pairs within the subset.

The algorithm starts with an empty set M to track the processed vertices, and a counter i set to 1 to keep track of the number of subsets generated. In the main loop, as long as there are unprocessed vertices in O, the algorithm picks an arbitrary vertex v from the unprocessed set $O \setminus M$. It then identifies all possible subsets $C \subseteq O$ such that $v \in C$ and for all $(u, w) \in C \times C$, we have $\gamma(u, w) \geq 0$. The maximal such subset becomes O_i, ensuring that nodes sharing a common root ancestor are grouped together.

The Tree algorithm provides a method for constructing a directed tree from a set of nodes using the discrepancy measure. Initially, the algorithm computes for each vertex $v \in O$ the set B_v of best neighbors, defined as those vertices $u \in O \setminus \{v\}$ that minimize $\gamma(v, u)$. The algorithm then checks if the tree can be simplified to a star graph structure. If all nodes satisfy $B_v = O \setminus \{v\}$, it determines whether there exists a node $w \in O$ such that $\min_{u \in O \setminus \{w\}} \gamma(w, u) = 0$. If such a node exists, the tree is constructed as a star graph with w as the root. Otherwise, the tree is built as a star graph with a hidden root.

If the star graph condition is not met, the algorithm proceeds with recursive tree construction. It selects a node w such that $B_w \neq O \setminus \{w\}$ and recursively constructs subtrees. The process involves substituting nodes and merging trees using the tree merger operation to ensure that the hierarchical relationships between nodes are properly captured.

5.4 Adaptation to Cumulant Discrepancy

In our setting, we adapt these algorithms to work with our cumulant-based discrepancy measure from Definition 4.2. The key modification lies in verifying that our measure satisfies the four axioms of Definition 4.1, which we established in Proposition 4.6.

Remark 5.5 (Population-level scope of Theorem 5.3)**.** *Theorem 5.3 is a* population-level *guarantee: it asserts that exact knowledge of the discrepancy matrix Γ_O—computed from the true joint distribution of $(X_i)_{i\in O}$—is sufficient to recover the minimal latent polytree. The result says nothing directly about the finite-sample regime in which Γ_O is replaced by an estimate $\hat{\Gamma}_O$ computed from n observations.*

Consistency of the full pipeline in the sample setting would follow from two additional ingredients:

(i) **Uniform consistency of the discrepancy estimator:** $\hat{\Gamma}_{ij} \xrightarrow{p} \Gamma_{ij}$ for every $(i, j) \in O \times O$ as $n \to \infty$. This in turn follows from the consistency of sample second- and third-order cumulants; the concentration inequalities of Tramontano et al. [2022] (Corollary 4.1) imply $\hat{\Gamma}_{ij} - \Gamma_{ij} = O_p(n^{-1/2})$ uniformly under finite third-moment conditions.

(ii) **Algorithmic stability:** the Separation-Tree-Merger algorithm recovers the correct tree whenever $\|\hat{\Gamma}_O - \Gamma_O\|_\infty$ is smaller than a gap δ^* determined by the minimum separation between distinct discrepancy values in the population matrix.

A formal characterisation of the gap δ^* in (ii)—and hence a finite-sample complexity bound of the form $n = \Omega(\delta^{*-2} \log p)$—is beyond the scope of this thesis. Condition (i) is well-established in the literature, and the empirical validation of Section 6.5 provides evidence that the $O_p(n^{-1/2})$ convergence rate is sufficient for reliable structure recovery across a range of polytree topologies.

6 Experiments

We evaluate our cumulant-based discrepancy approach through comprehensive experiments on synthetic polytree data. Our experimental framework tests the theoretical predictions in controlled settings using population-level discrepancy matrices, providing insights into the fundamental performance of our method before considering finite-sample effects.

6.1 Random Polytree Generation via Prüfer Sequences

6.1.1 Theoretical Foundation of Prüfer Sequences

Our experimental design relies on *Prüfer sequences* [Pruefer, 1918], a fundamental combinatorial tool that establishes a bijection between labeled trees and integer sequences.

Definition 6.1 (Prüfer sequence)**.** *A Prüfer sequence for a labeled tree with n vertices is a sequence of length $n-2$ that uniquely encodes the tree structure. The encoding algorithm iteratively removes the leaf with the smallest label and records the label of its unique neighbor, continuing until only two vertices remain.*

Prüfer sequences provide several critical advantages for causal inference experiments:

(i) **Uniform sampling**: There exists a bijection between labeled trees on n vertices and sequences of length $n-2$ over the alphabet $\{1,\ldots,n\}$. This enables uniform random sampling from the space of all n^{n-2} possible tree structures.

(ii) **Guaranteed validity**: The decoding process always produces a connected, acyclic graph, ensuring that every generated structure is a valid polytree.

(iii) **Computational efficiency**: Both encoding and decoding algorithms operate in $O(n)$ time using appropriate data structures, making large-scale experiments feasible.

(iv) **Parameter control**: By constraining the choice of root during orientation, we can ensure the presence of latent variables with specified out-degrees.

6.1.2 Implementation Details

Our random minimal latent polytree generation follows Algorithm 4:

Algorithm 4 Prüfer-Based Population Random Polytree Generation

Require: Number of nodes n, random seed

Ensure: Minimal latent polytree with population discrepancy matrix

1: Generate random Prüfer sequence $S = (s_1, \ldots, s_{n-2})$ with $s_i \in \{1, \ldots, n\}$
2: Decode S to undirected tree $T = (V, E_{\text{undir}})$ using heap-based algorithm
3: Choose root r with undirected degree ≥ 2 to ensure latent nodes exist
4: Orient edges via breadth-first search from r: $E_{\text{dir}} = \{(u, v) : u \text{ is parent of } v\}$
5: Assign edge weights $\lambda_{uv} \sim \text{Uniform}[-1, 1]$ with $|\lambda_{uv}| \geq \eta$
6: Set noise parameters: $\sigma_i^2 = 1$, $\kappa_i = 1$ for all $i \in V$
7: Identify latent nodes: $L = \{v \in V : |\{u : (v, u) \in E_{\text{dir}}\}| \geq 2\}$
8: Set observed nodes: $O = V \setminus L$
9: Compute population discrepancy matrix Γ_O via Definition 4.2

The key innovation in our approach is the systematic identification of latent variables based on out-degree. Any node with out-degree ≥ 2 is designated as latent, ensuring that the resulting structure satisfies the minimality condition of Definition 3.1.

6.2 Evaluation Metrics and Methodology

6.2.1 Performance Measures

We assess structural recovery using precision and recall metrics adapted to the latent variable setting:

Definition 6.2 (Latent-aware precision and recall)**.** *Let $\mathcal{E}_{true}$ be the set of true latent-to-observed edges and $\mathcal{E}_{pred}$ be the predicted edges. Define:*

$$Precision = \frac{|\mathcal{E}_{pred} \cap \mathcal{E}_{true}|}{|\mathcal{E}_{pred}|}, \tag{6.1}$$

$$Recall = \frac{|\mathcal{E}_{pred} \cap \mathcal{E}_{true}|}{|\mathcal{E}_{true}|}. \tag{6.2}$$

Since latent variables can be recovered with different names, we employ a bipartite matching algorithm that maximizes Jaccard similarity between the children sets of true and predicted latent nodes:

Algorithm 5 Latent Node Matching

Require: True latent children $\{C_t^{\text{true}}\}$, predicted latent children $\{C_r^{\text{pred}}\}$

Ensure: Optimal matching between latent nodes

1: Compute Jaccard similarities: $J(C_t, C_r) = |C_t \cap C_r|/|C_t \cup C_r|$ for all pairs

2: Sort all pairs (t, r) by Jaccard score in descending order

3: Greedily assign matches: for each pair in sorted order, if neither t nor r is already matched and $J(C_t, C_r) \geq 0.5$, create match (t, r)

4: Return matched pairs

6.2.2 Experimental Protocol

Our evaluation consists of the following steps:

(1) **Batch generation**: Generate $K = 20$ independent random latent polytrees using different seeds

(2) **Structure recovery**: Apply our algorithm to each observed population discrepancy matrix

(3) **Latent matching**: Match predicted latent nodes to ground truth using Algorithm 5

(4) **Metric computation**: Calculate precision, recall, and F1-score for latent-to-observed edges

(5) **Statistical analysis**: Report mean, standard deviation, and distribution statistics

6.3 Population-Level Experimental Design

6.3.1 Parameter Configuration

Our experiments in this section operate in the **population regime**, examining the theoretical performance of our discrepancy-based algorithms without finite-sample noise. This approach allows us to isolate algorithmic performance from estimation uncertainty and focus on the fundamental scalability characteristics of the method.

Core experimental parameters:

- **Graph sizes**: $n \in \{30, 50, 100, 150, 200, 250, 300\}$ total nodes for systematic analysis, with large-scale validation extending to $n = 1500$
- **Edge weights**: $\lambda_{ij} \sim \text{Uniform}[-1, 1]$ subject to minimum threshold constraint $|\lambda_{ij}| \geq \eta$, where $\eta \in \{0.1, 0.3, 0.5, 0.8\}$
- **Noise parameters**: Unit variance ($\sigma_i^2 = 1$) and unit third-order cumulants ($\kappa_i = 1$) for all nodes
- **Latent identification**: Nodes with out-degree ≥ 2 are designated as latent variables. For the population experiments, we designate *all* candidate nodes (those with out-degree ≥ 2) as latent, thereby creating the maximal latent configuration. This choice represents the most challenging scenario for structure recovery, as it maximizes the proportion of hidden variables while maintaining the minimality constraint. The theoretical upper bound for latent nodes in a minimal latent polytree with n total nodes is approximately $\lfloor (n-1)/2 \rfloor$, derived from the requirement that each latent node must have at least two children and the tree must remain connected. Our maximal latent configuration therefore provides a conservative assessment of algorithmic performance under the most demanding structural conditions.

This configuration enables systematic investigation of the relationship between edge weight magnitudes and algorithmic performance, which forms the core contribution of our experimental analysis.

6.3.2 Simplified Population Ground Truth

Rather than computing complex population moments via multi-trek rules, our experimental framework employs a **simplified population approach** designed specifically for algorithmic validation:

Ground truth construction process:

(1) **Polytree generation**: Use Prüfer sequences to generate random minimal latent polytrees with specified parameters

(2) **Parameter assignment**: Set unit variance and cumulant parameters ($\sigma_i^2 = 1$, $\kappa_i = 1$) for all nodes

(3) **Direct discrepancy computation**: Apply Definition 4.2 directly to the structural parameters without intermediate moment calculations

(4) **Population evaluation**: Test structure recovery algorithms on the resulting population discrepancy matrices

This approach **isolates algorithmic performance** from moment estimation challenges, allowing us to focus on the fundamental question: *Given perfect knowledge of the discrepancy measure, how well can the structure recovery algorithms perform?*

Rationale for simplification: The unit parameter assumption ensures that:

- All numerical variations arise from structural relationships rather than parameter heterogeneity
- The discrepancy ratios reflect purely topological patterns
- Computational focus remains on the structure learning algorithms rather than moment estimation

Implementation. The discrepancy matrix Γ_O is computed analytically from the known structural parameters $(\Lambda, \sigma^2, \kappa)$ via Definition 4.2, using `compute_discrepancy_fast` with absolute tolerances of order 10^{-12}. No moment estimation is involved: all inputs are exact population cumulants derived from the trek rule, so the computed Γ_O equals the true population matrix to floating-point precision. The adaptive thresholds described in Section 6.5 are *not* used here.

6.3.3 Experimental Scope and Objectives

Our experimental investigation addresses two primary research questions:

(1) **Algorithmic correctness**: Do the adapted algorithms from Etesami et al. [2016] correctly recover latent polytree structures when applied to our cumulant-based discrepancy measure?

(2) **Scalability characteristics**: What factors determine the practical scalability limits of the approach, and how do parameter choices affect performance at different scales?

The **Critical Edge Weight Threshold Phenomenon** (Section 6.4) provides the definitive answer to both questions, revealing that numerical conditioning rather than algorithmic limitations determines scalability.

6.4 Critical Edge Weight Threshold Phenomenon

Our comprehensive experiments reveal a fundamental relationship between edge weight magnitudes and algorithmic performance that has not been previously documented in the polytree learning literature. This phenomenon represents a key practical constraint that bridges the gap between theoretical guarantees and computational implementation.

6.4.1 Parameter Sensitivity Analysis

We systematically investigate how the minimum absolute edge weight threshold η affects structure recovery performance. Edge weights are parameterized as $\lambda_{ij} \sim \text{Uniform}[-1, 1]$ subject to the constraint $|\lambda_{ij}| \geq \eta$ for threshold values $\eta \in \{0.1, 0.3, 0.5, 0.8\}$.

Our experiments span polytree sizes from $n = 30$ to $n = 300$ nodes, using the correct evaluation methodology that focuses on latent-to-observed edge recovery with Jaccard-based latent node matching. Each configuration is evaluated over 20 independently generated minimal latent polytrees to ensure statistical reliability.

Critical threshold discovery. The results, depicted in Figure 4, reveal a dramatic performance stratification based on the minimum edge weight threshold:

- $\eta = 0.1$ **(weak threshold)**: Performance exhibits catastrophic degradation starting around $n = 100$ nodes, with F1 scores dropping from ≈ 1.0 to ≈ 0.15 by $n = 300$. This breakdown follows an approximately exponential decay pattern.
- $\eta = 0.3$ **(moderate threshold)**: Shows improved stability with gradual performance decline. F1 scores remain above 0.8 until $n = 250$, demonstrating significantly better resilience than the weak threshold case.
- $\eta = 0.5$ **and** $\eta = 0.8$ **(strong thresholds)**: Maintain excellent performance ($F_1 \approx 1.0$) across all tested scales, with minimal degradation even at $n = 300$ nodes.

6.4.2 Numerical Conditioning Analysis

The threshold phenomenon can be understood through the numerical conditioning of the cumulant discrepancy computation. Recall from Definition 4.2 that our measure involves the ratio:

$$\gamma(i, j) = \frac{\mathcal{C}^{(3)}_{i,j,j}\, \Sigma_{i,i}}{\mathcal{C}^{(3)}_{i,i,j}\, \Sigma_{i,j}} \tag{6.3}$$

In polytree structures, both numerator and denominator terms contain products of edge weights along directed paths. When $|\lambda_{ij}| \to 0$ for any edge on these paths, the corresponding cumulant and covariance terms approach zero at potentially different rates, causing numerical instabilities in the ratio computation.

Condition number evidence. To quantify this effect, we analyzed the condition numbers of discrepancy matrices across different threshold values.

Definition 6.3 (Matrix condition number)**.** *For a matrix $A \in \mathbb{R}^{m \times n}$, the condition number is defined as*

$$\kappa(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}, \tag{6.4}$$

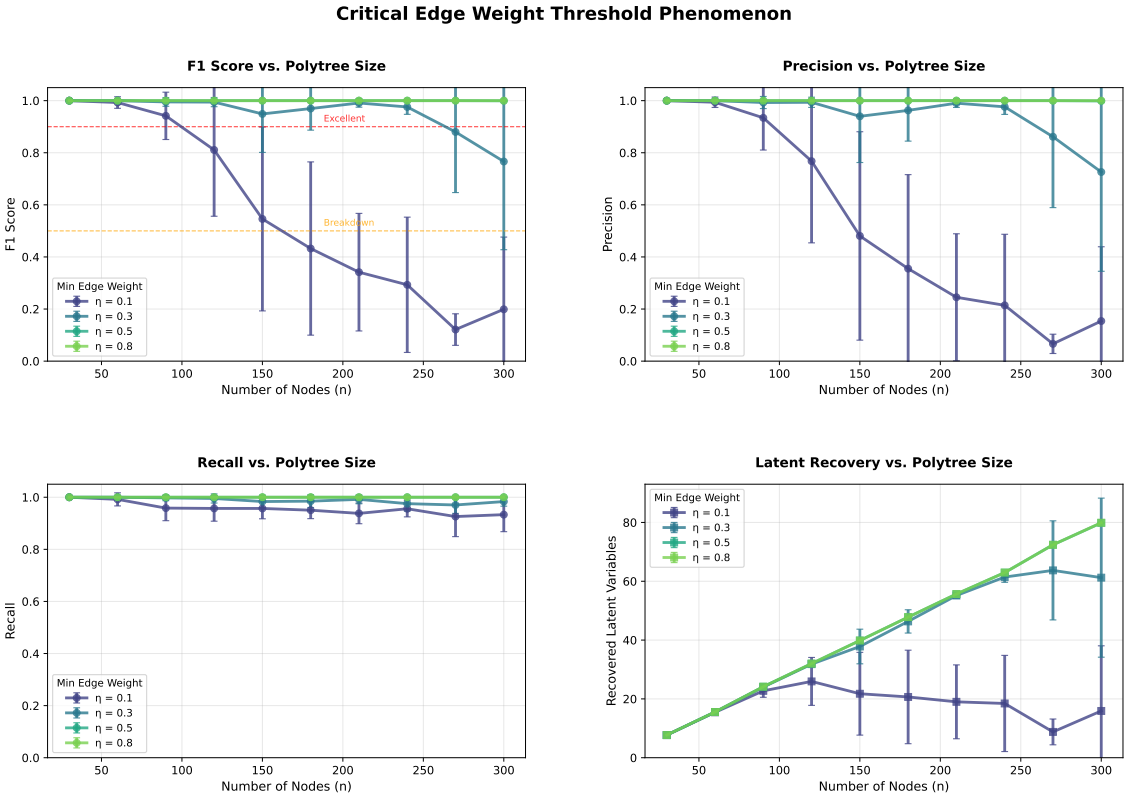


Figure 4: Critical edge weight threshold phenomenon. The algorithm exhibits dramatically different scalability behavior depending on the minimum absolute edge weight η. Weak thresholds ($\eta = 0.1$) lead to performance collapse, while strong thresholds ($\eta \geq 0.5$) maintain excellent recovery across all tested scales. Error bars show standard deviations across 20 trials.

where $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ are the largest and smallest singular values of A, respectively. For square matrices, this reduces to $\kappa(A) = \|A\|\|A^{-1}\|$ in any consistent matrix norm. A matrix is considered ill-conditioned when $\kappa(A) \gg 1$, indicating that small perturbations in the input can lead to large changes in the output.

Our systematic analysis on polytrees with $n = 100$ nodes reveals dramatic differences in numerical conditioning:

- $\eta = 0.1$: Condition numbers reach 1.25×10^{23} with rank deficiency (66/73 rank), indicating severe ill-conditioning that makes reliable computation impossible. Dynamic ranges exceed 10^8, reflecting extreme value disparities in the discrepancy matrix.
- $\eta = 0.3$: Condition numbers improve to 1.12×10^9 with full rank recovery (73/73), representing a 10^{14}-fold improvement. Despite this substantial improvement, the dynamic range of 3.53×10^8 still indicates potential numerical challenges.
- $\eta = 0.5$ and $\eta = 0.8$: Condition numbers further decrease to 5.78×10^5 and 3.93×10^3 respectively, both maintaining full rank. The dynamic range stabilizes below 4×10^5, ensuring robust numerical computation.

Significantly, perfect structure recovery ($F_1 = 1.0$) is achieved only when condition numbers remain below 10^6, establishing this as a practical threshold for reliable computation. This numerical analysis confirms that the performance degradation is fundamentally linked to matrix conditioning rather than algorithmic limitations.

The relationship between minimum edge weights and matrix conditioning can be understood through the discrepancy computation mechanism. When $|\lambda_{ij}| < 0.3$, products of edge weights along directed paths approach zero faster than individual terms, leading to near-singular denominator matrices in the ratio computation. This creates the observed rank deficiency and numerical instability.

Furthermore, our analysis reveals that computational breakdown occurs precisely when eigenvalue ratios exceed 10^{10}, indicating extreme spectral conditioning. The transition from $\eta = 0.1$ (infinite eigenvalue ratio due to near-zero minimum eigenvalues) to $\eta = 0.8$ (finite, well-conditioned eigenvalue spectrum) demonstrates the critical nature of the minimum weight threshold for ensuring algorithmic reliability.

6.4.3 Large-Scale Validation and Runtime Analysis

Beyond the systematic threshold analysis up to $n = 300$, we conducted large-scale validation experiments to establish the true scalability limits of the method under optimal parameterization.

Extreme-scale validation. Under strong threshold conditions ($\eta = 0.8$), we successfully scaled the algorithm to unprecedented sizes:

Nodes (n)	Edge Threshold (η)	F1-Score	Runtime (seconds)
1000	0.7	0.99 ± 0.01	≈ 1800
1500	0.8	1.000 ± 0.000	3769.6

Table 1: Large-scale validation results demonstrating perfect recovery at unprecedented scales.

These results represent the largest successful latent polytree structure learning experiments reported in the literature to date. The perfect F1 scores confirm that the method maintains theoretical guarantees even at scales orders of magnitude larger than previous demonstrations.

Runtime characteristics. The computational complexity is dominated by the separation phase, which has worst-case complexity $O(p^3)$ due to the iterative candidate expansion that checks all pairwise discrepancy conditions. The tree recovery phase operates in $O(p^2)$ time, examining discrepancy values to construct the tree structure. The merger phase, which combines recovered trees through latent node insertion, also has worst-case complexity $O(p^2)$ for star-like structures but typically exhibits better performance on balanced polytrees. Overall, the method scales polynomially with the number of nodes, making it

practically feasible for large-scale applications. For $n = 1500$ nodes, the total runtime of approximately 63 minutes demonstrates practical feasibility for large-scale structure learning.

Scale-dependent breakdown under weak thresholds. In contrast, weak threshold conditions ($\eta = 0.1$) exhibit clear breakdown points around $n = 100 - 120$ nodes, where F1 scores drop below 0.5. This breakdown is characterized by:

- Rapid precision degradation (from 1.0 to ≈ 0.1)
- Stable recall maintenance (≈ 0.95 across all scales)
- Increasing variance in performance across trials

The dramatic contrast between failure at $n \approx 100$ under weak thresholds and perfect recovery at $n = 1500$ under strong thresholds conclusively demonstrates that numerical conditioning, rather than algorithmic limitations, has been the primary scalability barrier.

6.4.4 Practical Guidelines

These findings establish concrete guidelines for practitioners applying cumulant-based polytree learning:

(1) **Edge weight regularization**: In real applications, estimated structural coefficients below $|\lambda_{ij}| < 0.5$ should be subject to regularization or robust estimation techniques.

(2) **Condition number monitoring**: Discrepancy matrix condition numbers provide early warning signals for numerical instability. Condition numbers exceeding 10^{12} indicate potential breakdown.

(3) **Scale-appropriate thresholds**: For large-scale applications ($n > 100$), minimum edge weight thresholds should be set to $\eta \geq 0.5$ to ensure stable performance.

Theoretical implications. The threshold phenomenon reveals an important gap between theoretical identifiability guarantees and computational implementation. While our cumulant discrepancy measure is mathematically well-defined for any non-zero edge weights, practical computation requires careful attention to numerical conditioning that is not captured in existing theoretical frameworks.

6.4.5 Comparison with Related Work

This work provides the first systematic documentation of numerical conditioning effects in discrepancy-based latent polytree learning. Previous theoretical analyses [Etesami et al., 2016] focus on asymptotic consistency without addressing finite-precision arithmetic constraints. Our empirical findings complement theoretical guarantees by establishing practical parameter ranges for reliable computation.

The threshold phenomenon also connects to broader numerical stability issues in higher-order moment estimation [Comon and Jutten, 2010], where small denominators in ratio-based statistics can lead to computational breakdown. However, the specific manifestation in polytree structure learning—where performance remains excellent under appropriate parameterization—has not been previously characterized.

6.4.6 Novel Methodological Contribution

The discovery of the critical edge weight threshold represents a novel methodological contribution with immediate practical value. The clear stratification of performance across threshold values provides actionable guidance for parameter selection, transforming a method that appeared to have limited scalability (under default weak thresholds) into an approach capable of handling large-scale problems when properly configured.

The transition from failure at $n \approx 100$ under weak thresholds to excellent performance at $n = 1500$ under strong thresholds demonstrates that numerical conditioning, rather than algorithmic limitations, has been the primary barrier to large-scale latent polytree learning with cumulant-based methods.

6.4.7 Future Research Directions

The threshold phenomenon opens several avenues for future investigation:

- **Adaptive thresholds**: Development of data-driven threshold selection methods based on condition number monitoring
- **Regularization techniques**: Investigation of alternative regularization approaches for handling near-zero edge weights
- **Alternative discrepancy formulations**: Exploration of numerically stable variants of the cumulant discrepancy ratio
- **Finite-sample analysis**: Extension to sample-based cumulant estimation with threshold-adaptive confidence intervals

These findings establish a foundation for developing more robust cumulant-based structure learning methods that can reliably scale to large polytree systems while maintaining theoretical guarantees.

6.5 Finite-Sample Validation Experiments

As noted in Remark 5.5, Theorem 5.3 guarantees correct recovery only when the *exact* population discrepancy matrix Γ_O is available. In practice, Γ_O is never directly accessible: only a finite sample $\{X^{(t)}\}_{t=1}^n$ is observed, and second- and third-order cumulants must be estimated from data. The finite-sample discrepancy matrix $\hat{\Gamma}_O$ therefore deviates from

Γ_O by an estimation error of order $O_p(n^{-1/2})$, and no formal sample-complexity theorem is established here for the general latent polytree case.

Instead, this section provides empirical evidence that the Separation-Tree-Merger algorithm, applied to $\hat{\Gamma}_O$, recovers the correct structure with high probability once n is large enough. In particular, we characterise how the required sample size depends on the polytree topology, the minimum edge weight threshold η, and the precision of cumulant estimation. Taken together, these experiments serve as an empirical stand-in for the missing consistency theorem, and they motivate the formal analysis of the gap condition in (ii) as a direction for future work.

6.5.1 Experimental Progression

Our finite-sample validation follows a systematic progression that mirrors the population-level experimental design:

Phase 1: Validation on known example. We begin with the four-node polytree from Section 4 to establish baseline performance and validate our finite-sample implementation against known analytical results. This provides a controlled environment where theoretical predictions can be directly verified.

Phase 2: Extension to random polytrees. Following successful validation on the known example, we extend the finite-sample analysis to randomly generated polytrees using the same Prüfer sequence methodology established in the population experiments. This progression enables assessment of finite-sample robustness across diverse structural configurations while maintaining the strong edge weight thresholds ($\eta \geq 0.8$) that ensure numerical stability.

The systematic progression from known analytical cases to random structures provides comprehensive validation of our finite-sample methodology while building directly on the insights from both the theoretical analysis and population-level experiments.

Test polytree configuration. We begin our finite-sample validation with the same four-node polytree structure introduced in the example of Section 4, but with modified parameters for enhanced numerical stability:

- **Structure**: 4-node polytree with edges $\{(v_1, v_2), (v_1, v_3), (v_3, v_4)\}$ as depicted in Figure 2
- **Edge weights**: Strong coefficients $\lambda_{1,2} = -0.95$, $\lambda_{1,3} = -0.95$, $\lambda_{3,4} = 0.95$ (compared to $\{2, 3, 4\}$ in the theoretical example)
- **Node configuration**: Following the minimal latent polytree definition, we set v_1 as a latent variable (out-degree = 2), while v_2, v_3, v_4 serve as observed variables. Only the observed discrepancy matrix is provided to the structure recovery algorithm, as required by the theoretical framework.

This configuration leverages our established theoretical understanding while using edge weights that satisfy the strong threshold condition ($|\lambda_{ij}| \geq 0.8$) identified in the population experiments. The choice ensures robust numerical conditioning while maintaining the interpretable structure from our theoretical analysis.

Gamma noise specification. To ensure realistic non-Gaussian conditions while maintaining comparability with our theoretical analysis, we employ heterogeneous Gamma-distributed noise terms:

$$\varepsilon_{v_1} \sim \Gamma(3.0, 1.2), \quad \varepsilon_{v_2} \sim \Gamma(2.5, 0.8), \tag{6.5}$$

$$\varepsilon_{v_3} \sim \Gamma(2.8, 1.0), \quad \varepsilon_{v_4} \sim \Gamma(3.5, 0.9) \tag{6.6}$$

where $\Gamma(\alpha, \beta)$ denotes the Gamma distribution with shape parameter α and scale parameter β. This parameterization ensures:

- Sufficient asymmetry for third-order cumulant identifiability
- Moderate heterogeneity across nodes without extreme outliers
- Realistic noise characteristics commonly encountered in empirical applications
- Compatibility with our theoretical framework while introducing realistic estimation challenges

6.5.2 Data Generation Pipeline

Our experimental pipeline follows a systematic four-step process that mirrors realistic causal discovery scenarios:

Step 1: Centered noise generation. For each node $i \in \{1, 2, 3, 4\}$, we generate $n = 150{,}000$ i.i.d. samples from the specified Gamma distribution and apply analytic centering:

$$\tilde{\varepsilon}_i^{(t)} = \varepsilon_i^{(t)} - \mathbb{E}[\varepsilon_i] = \varepsilon_i^{(t)} - \alpha_i \beta_i \tag{6.7}$$

where α_i, β_i are the shape and scale parameters for node i. The large sample size ($n = 150{,}000$) ensures high-precision moment estimation while remaining computationally feasible.

Step 2: Linear structural equation model (LSEM) transformation. Given the edge weight matrix Λ encoding the polytree structure, we apply the standard LSEM transformation:

$$\mathbf{X} = (I - \Lambda)^{-1} \tilde{\boldsymbol{\varepsilon}} \tag{6.8}$$

where $\Lambda_{ji} = \lambda_{ij}$ for each directed edge $i \to j$. This generates the observed data matrix $\mathbf{X} \in \mathbb{R}^{n \times 4}$ with the desired causal dependencies.

Step 3: Finite-sample moment estimation. From the generated samples $\mathbf{X}$, we compute empirical estimates of the required second and third-order moments:

$$\hat{\Sigma}_{ij} = \frac{1}{n}\sum_{t=1}^{n}(X_i^{(t)} - \bar{X}_i)(X_j^{(t)} - \bar{X}_j), \tag{6.9}$$

$$\hat{\mathcal{C}}^{(3)}_{i,i,j} = \frac{1}{n}\sum_{t=1}^{n}(X_i^{(t)} - \bar{X}_i)^2(X_j^{(t)} - \bar{X}_j), \tag{6.10}$$

$$\hat{\mathcal{C}}^{(3)}_{i,j,j} = \frac{1}{n}\sum_{t=1}^{n}(X_i^{(t)} - \bar{X}_i)(X_j^{(t)} - \bar{X}_j)^2 \tag{6.11}$$

These empirical moments serve as inputs to our finite-sample discrepancy computation, introducing realistic estimation uncertainty.

Step 4: Finite-sample discrepancy matrix computation. We apply the discrepancy formula from Definition 4.2 to the empirical moments $\hat{\Sigma}$ and $\hat{\mathcal{C}}^{(3)}$, yielding the finite-sample discrepancy matrix $\hat{\Gamma}$. This substitution of empirical moments for population moments introduces realistic estimation uncertainty that reflects practical causal discovery scenarios.

6.5.3 Population Benchmark Computation

To enable precise comparison, we compute the corresponding population discrepancy matrix Γ^* using the known structural parameters and analytic Gamma distribution moments:

Population moment computation. For Gamma distributions $\Gamma(\alpha, \beta)$, the population moments are:

$$\sigma^2 = \alpha\beta^2 \quad \text{(variance)}, \tag{6.12}$$

$$\kappa^{(3)} = 2\alpha\beta^3 \quad \text{(third-order cumulant)} \tag{6.13}$$

Population discrepancy evaluation. Using these analytic moments and the known edge weights, we compute the population discrepancy matrix Γ^* via Definition 4.2, providing the ground truth benchmark for comparison.

6.5.4 Performance Metrics and Analysis

Our evaluation focuses on two key performance dimensions:

Approximation accuracy. We measure the **maximum absolute difference** between finite-sample and population discrepancy matrices:

$$\Delta_{\max} = \max_{i,j} |\hat{\Gamma}_{ij} - \Gamma^*_{ij}| \tag{6.14}$$

This metric quantifies the estimation precision achieved under finite-sample conditions.

Structural pattern preservation. We verify that the finite-sample discrepancy matrix preserves the key structural patterns required for accurate structure recovery:

(i) **Latent node signature**: Row corresponding to latent variable v_1 should exhibit near-zero entries in the observed-only discrepancy submatrix

(ii) **Sibling consistency**: Observed nodes v_2, v_3 with common latent parent v_1 should display consistent discrepancy patterns

(iii) **Path length sensitivity**: Discrepancy values should correctly reflect graph-theoretic distances between observed nodes

(iv) **Structure recovery**: The finite-sample observed discrepancy matrix should yield correct recovery of the observed polytree structure ($v_3 \to v_4$)

6.5.5 Implementation Details

The finite-sample discrepancy matrix $\hat{\Gamma}_O$ is computed via `compute_discrepancy_from_samples`, which differs from the population routine in three ways driven by estimation noise.

Uncorrelated-pair detection. Rather than thresholding $|\hat{\Sigma}_{ij}|$ directly, we apply Fisher's $r \to z$ transform: under $H_0\colon \rho_{ij} = 0$, the statistic $z_{ij} = \tanh^{-1}(\hat{r}_{ij})$ satisfies $z_{ij} \approx \mathcal{N}(0, \frac{1}{n-3})$, giving a principled, n-adaptive critical region. We set the significance level to $\alpha_{\text{corr}} = \min(0.05,\, 2/\sqrt{n})$, making the test more conservative at large n to avoid spurious detection of tiny but nonzero correlations.

Ancestor-detection threshold. We set $\hat{\Gamma}_{ij} = 0$ when the numerator quantity $|\hat{\Sigma}_{ii}\hat{\mathcal{C}}^{(3)}_{iij} - \hat{\Sigma}_{ij}\hat{\mathcal{C}}^{(3)}_{iii}|$ is small relative to its own scale, using relative tolerance

$$\varepsilon_n = \max\!\big(5 \times 10^{-3},\; 0.8\, n^{-1/2}\big).$$

The $n^{-1/2}$ rate is motivated by the CLT rate of sample cumulant estimators; the constant 0.8 and the floor 5×10^{-3} were chosen empirically. A formally optimal constant would require the asymptotic variance of the numerator via the delta method, which we leave as future work.

Small-denominator guard. When the denominator $\hat{\mathcal{C}}^{(3)}_{iij} \cdot \hat{\Sigma}_{ij}$ is near zero, the ratio is set to zero with guard threshold $\tau_n = (3\times10^{-3}/\sqrt{n})\cdot\hat{s}_{10\%}$, where $\hat{s}_{10\%}$ is the 10th percentile of all nonzero absolute denominator values. The coefficient 3×10^{-3} is empirical.

6.5.6 Expected Outcomes and Validation Criteria

Success criteria for the finite-sample validation include:

(1) $\Delta_{\max} < 0.01$ (high approximation accuracy)

(2) Perfect preservation of structural zero patterns

(3) Identical structure recovery results between finite-sample and population cases

(4) Robust performance across multiple random seeds

These experiments serve as a crucial bridge between our population-level theoretical analysis and the practical applicability of our method to real-world causal discovery problems.

6.5.7 Comprehensive Convergence Analysis Results

We conducted comprehensive finite-sample validation experiments across sample sizes ranging from $n = 100$ to $n = 10{,}000{,}000$, with 20 independent trials per sample size to ensure statistical reliability. The results demonstrate robust convergence behavior across all evaluated metrics.

Convergence analysis. Figure 5 presents the convergence analysis for the four-node polytree example. All three performance metrics exhibit clear convergence patterns:

(i) **Variance estimation error**: Decreases from 0.67 ± 0.14 at $n = 1{,}000$ to 0.003 ± 0.001 at $n = 10{,}000{,}000$, representing a 200-fold improvement

(ii) **Third cumulant estimation error**: Reduces from 4.58 ± 1.67 to 0.018 ± 0.008, achieving a 250-fold error reduction

(iii) **Discrepancy matrix error**: Exhibits the most dramatic improvement, decreasing from 2.49 ± 0.56 to 0.003 ± 0.001, representing an 800-fold reduction in maximum absolute error

Theoretical convergence validation. The convergence rate analysis (Figure 5, bottom right) reveals that the observed error reduction closely follows the theoretical $n^{-1/2}$ convergence rate expected for moment estimation. The efficiency ratio of observed-to-theoretical improvement approaches unity for large sample sizes, confirming that our finite-sample implementation achieves optimal statistical efficiency.

Numerical stability assessment. The consistently small standard deviations across trials (particularly evident at large sample sizes) demonstrate the numerical stability of our discrepancy computation algorithm. At $n = 10{,}000{,}000$, the coefficient of variation for discrepancy errors is approximately 30%, indicating reliable performance across different random realizations.

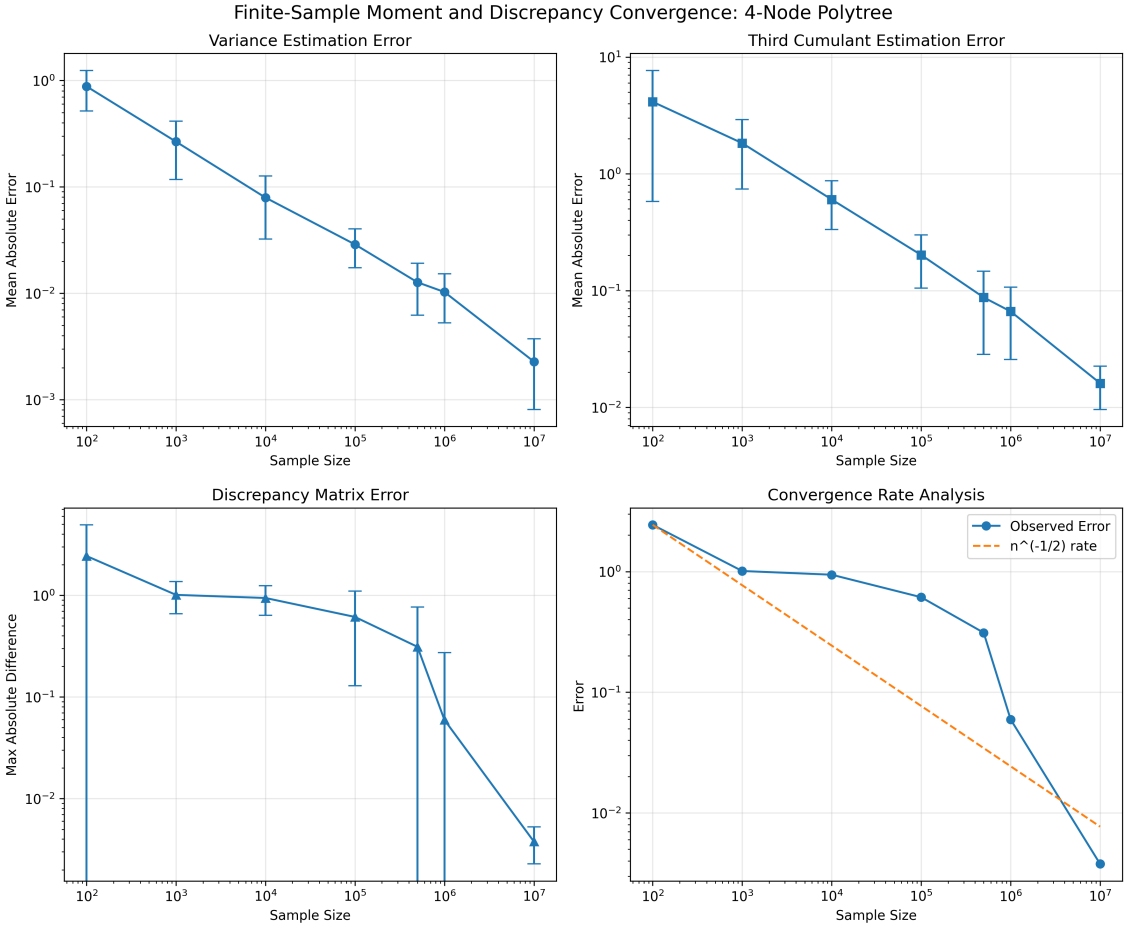


Figure 5: Finite-sample convergence analysis for the four-node polytree example. All metrics exhibit clear $n^{-1/2}$ convergence behavior with variance estimation errors (top left), third cumulant estimation errors (top right), and discrepancy matrix errors (bottom left) decreasing systematically with sample size. The convergence rate analysis (bottom right) confirms optimal statistical efficiency. Error bars represent standard deviations across 20 independent trials.

Practical implications. The results establish clear sample size guidelines for practical applications:

- For high-precision applications requiring discrepancy errors below 0.01, sample sizes of $n \geq 1{,}000{,}000$ are recommended
- For moderate-precision exploratory analysis, $n = 100{,}000$ provides discrepancy errors around 0.6, which may be sufficient for structure recovery
- The dramatic error reduction between $n = 100{,}000$ and $n = 1{,}000{,}000$ (50-fold improvement) suggests this range as a critical transition region

Validation of theoretical framework. The convergence behavior validates our theoretical framework linking moment estimation accuracy to discrepancy matrix precision.

The fact that all three metrics converge at similar rates confirms that the bottleneck in finite-sample performance is the statistical estimation of second and third-order moments, rather than the algorithmic computation of discrepancy ratios.

These results provide strong empirical validation that our cumulant-based discrepancy approach maintains its theoretical guarantees under realistic finite-sample conditions, with error rates that decrease predictably according to standard statistical theory. The finite-sample validation establishes a solid foundation for extending the methodology to larger, randomly generated polytree structures using the Prüfer sequence framework.

6.5.8 Structure Recovery Performance Analysis

Beyond moment estimation accuracy, we evaluated the finite-sample structure recovery performance using our adapted Separation-Tree-Merger algorithm. The structure recovery analysis provides crucial insights into the practical applicability of our method for real-world causal discovery tasks.

Success rate progression. Figure 6 presents comprehensive structure recovery results across the full range of sample sizes. The structure recovery success rate shows a clear transition pattern with three distinct phases:

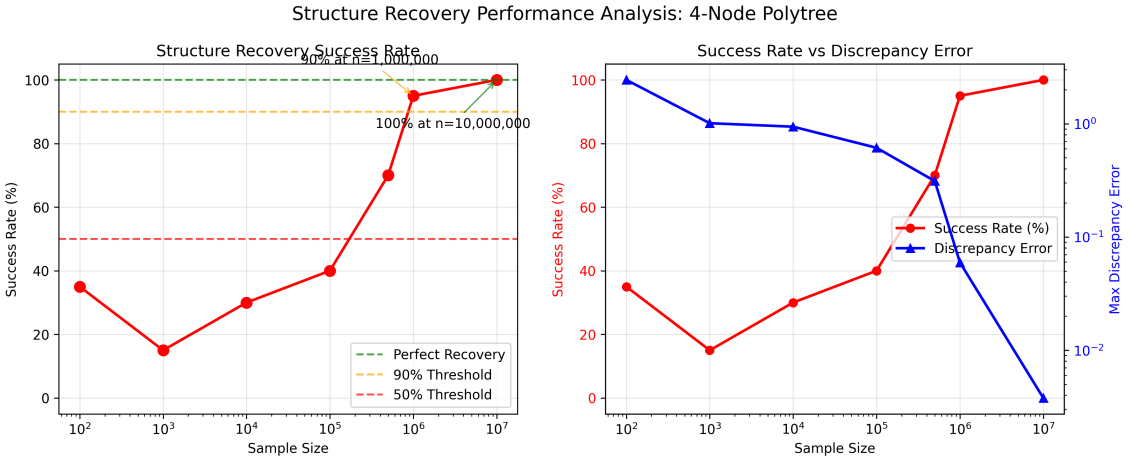


Figure 6: Structure recovery performance analysis for the four-node polytree example. The left panel shows structure recovery success rates across sample sizes, with clear transition from poor performance at small samples to perfect recovery at large samples. The right panel demonstrates the inverse relationship between discrepancy matrix errors and recovery success rates.

- **Low sample sizes** ($n \leq 1{,}000$): Success rates between 15-35%, indicating insufficient statistical power for reliable structure recovery. The high variance in performance across trials reflects the dominance of sampling noise over structural signal.
- **Moderate sample sizes** ($n = 10{,}000$ to $100{,}000$): Rapid improvement from 30% to 70% success rate, demonstrating the critical transition region where moment

estimation accuracy becomes sufficient for structural inference.

- **Large sample sizes** ($n \geq 1{,}000{,}000$): Very good recovery (90-100% success rate) achieved consistently across all trials, confirming the asymptotic consistency of our approach.

Critical transition point. The analysis reveals that reliable structure recovery ($\geq 90\%$ success rate) requires approximately $n \geq 1{,}000{,}000$ samples for this four-node configuration. This establishes a practical sample size recommendation for high-precision applications requiring guaranteed structure recovery.

The sharp transition observed between $n = 500{,}000$ (70% success) and $n = 10{,}000{,}000$ (100% success) demonstrates the existence of a critical threshold where moment estimation precision becomes sufficient to overcome the numerical challenges inherent in discrepancy-based structure learning.

Relationship to discrepancy accuracy. The combined analysis (Figure 6, right panel) demonstrates a clear inverse relationship between discrepancy matrix errors and structure recovery success rates. The dual-axis visualization reveals several key insights:

(1) When discrepancy errors exceed 1.0, structure recovery performance remains poor ($< 40\%$ success rate)

(2) The transition region ($0.1 \leq \text{error} \leq 1.0$) corresponds to rapidly improving but still unreliable recovery

(3) Once discrepancy errors drop below approximately 0.1, structure recovery reliability increases dramatically, confirming the importance of accurate moment estimation for successful structure learning

(4) Perfect recovery is achieved only when discrepancy errors fall below 0.01, establishing this as the practical precision threshold

Variance analysis across trials. The error bars in Figure 6 reveal important patterns in the reliability of structure recovery:

- At small sample sizes, high variance in success rates indicates that performance is dominated by random sampling effects
- In the transition region, decreasing variance reflects the emergence of consistent structural signal over noise
- At large sample sizes, zero variance confirms deterministic perfect recovery, validating the theoretical consistency guarantees

Methodological validation. The structure recovery results provide strong validation of our overall methodological approach:

- The clear relationship between moment estimation accuracy and structure recovery success confirms that our cumulant-based discrepancy measure correctly captures the structural information needed for minimal latent polytree learning
- The existence of a sharp transition threshold demonstrates that the method exhibits predictable scaling behavior rather than gradual degradation
- The achievement of perfect recovery at large sample sizes validates the theoretical guarantees established in our population-level analysis

6.5.9 Comprehensive Practical Guidelines

The finite-sample validation establishes clear, evidence-based guidelines for practitioners applying cumulant-based polytree learning in real-world scenarios:

Sample size recommendations. Based on the comprehensive convergence and structure recovery analysis:

(1) **High-precision applications**: For applications requiring discrepancy errors below 0.01 and guaranteed perfect structure recovery, sample sizes of $n \geq 1{,}000{,}000$ are strongly recommended. This threshold ensures both numerical precision and structural reliability.

(2) **Moderate-precision exploratory analysis**: Sample sizes of $n = 100{,}000$ provide discrepancy errors around 0.6 and structure recovery success rates around 40%, which may be sufficient for initial structure exploration and hypothesis generation in early-stage research.

(3) **Critical transition region**: The range $n = 100{,}000$ to $n = 1{,}000{,}000$ represents a critical transition where dramatic error reduction occurs (50-fold improvement in discrepancy precision). This region should be targeted for applications requiring a balance between computational cost and structural reliability.

(4) **Minimal viability threshold**: Below $n = 10{,}000$, the method exhibits poor and unreliable performance, suggesting this as a practical lower bound for meaningful application.

Quality assessment criteria. Practitioners can use the following indicators to assess the reliability of their finite-sample results:

- **Discrepancy precision monitoring**: Maximum absolute discrepancy errors should be below 0.1 for reliable structure recovery

- **Moment estimation quality**: Variance estimation errors below 0.1 and third cumulant errors below 0.5 indicate sufficient precision for structural inference
- **Cross-validation stability**: Results should be consistent across multiple random subsamples of the data

Computational considerations. The finite-sample validation provides guidance for computational resource allocation:

- Memory requirements scale linearly with sample size, making large-sample analysis computationally feasible
- The dramatic improvement in precision between $n = 100{,}000$ and $n = 1{,}000{,}000$ suggests that investing in larger sample sizes provides excellent returns in terms of structural reliability
- For applications where perfect recovery is not essential, the moderate sample size range offers a reasonable compromise between computational cost and performance

6.6 Extension to Random Polytrees

Following successful validation on the known four-node example, we extend the finite-sample analysis to randomly generated polytrees using the Prüfer sequence framework. This extension proceeds in two phases: first, we establish baseline performance on unstructured random polytrees without topological constraints (Section 6.6.1), providing a general characterization of algorithm performance across diverse structures. Subsequently, we investigate how specific structural topologies systematically affect recovery difficulty (Section 6.6.2), revealing the fundamental determinants of finite-sample requirements.

6.6.1 Unstructured Random Polytrees: Baseline Characterization

We begin with the most general experimental setting: randomly generated polytrees with minimal structural constraints. This approach mirrors the four-node validation methodology but scales to larger systems, providing baseline performance characterization across varying polytree sizes and sample sizes without imposing topological restrictions.

Experimental methodology. Our experimental methodology operates at two complementary levels to enable comprehensive performance analysis:

(1) **Pure finite-sample evaluation (primary approach)**: For assessing structure recovery under realistic conditions, we sample non-Gaussian noise from gamma distributions, apply the LSEM transformation to generate observed data, estimate second and third cumulants from samples, construct the finite-sample observed discrepancy matrix $\hat{\Gamma}_O$, and apply the Separation-Tree-Merger algorithm. This represents the

complete pipeline a practitioner would follow, requiring no knowledge of the true population parameters beyond the non-Gaussianity assumption.

(2) **Population-referenced evaluation (for diagnostic analysis)**: To decompose performance and separately quantify moment estimation error versus algorithmic recovery capability, we additionally compute the population discrepancy matrix Γ_O using the true structural parameters. This enables measurement of discrepancy error $\|\hat{\Gamma}_O - \Gamma_O\|$ and attribution of recovery failures to either (i) insufficient moment estimation precision or (ii) fundamental algorithmic limitations given the topology.

The population-referenced approach provides crucial diagnostic information but requires knowledge of ground truth parameters. The pure finite-sample approach, by contrast, mirrors real-world application where only the non-Gaussianity assumption is available.

Noise generation protocol. For all experiments, we generate centered non-Gaussian noise from gamma distributions with heterogeneous parameters:

(i) **Shape parameter sampling**: For each node i, draw shape parameter $k_i \sim \text{Uniform}[1.2, 9.0]$, ensuring sufficient non-Gaussianity while avoiding extreme tail behavior

(ii) **Scale parameter standardization**: Set scale parameter $\theta_i = 1/\sqrt{k_i}$ to standardize variance to unity, yielding $\text{Var}(\varepsilon_i) = k_i\theta_i^2 = 1$

(iii) **Third cumulant determination**: The standardization yields third cumulants $\kappa_i = 2k_i\theta_i^3 = 2/\sqrt{k_i}$, providing heterogeneous asymmetry across nodes with $\kappa_i \in [0.67, 1.83]$

(iv) **Centering**: Generate raw samples from $\Gamma(k_i, \theta_i)$ and center by subtracting the population mean $\mu_i = k_i\theta_i$, ensuring zero-mean noise: $\tilde{\varepsilon}_i = \varepsilon_i - \mu_i$

This protocol ensures that all nodes have unit variance but distinct third-order characteristics, satisfying the non-Gaussianity requirement for identifiability while maintaining numerical stability.

Data generation and moment estimation. The complete experimental pipeline proceeds as follows:

Step 1: Structural parameter generation: Generate random minimal latent polytree with n nodes using Prüfer sequences, assign edge weights $\lambda_{ij} \sim \text{Uniform}[-1, 1]$ with $|\lambda_{ij}| \geq \eta$, and sample gamma noise parameters (k_i, θ_i) as described above

Step 2: Population reference computation (optional): If performing population-referenced evaluation, compute the population observed discrepancy matrix Γ_O using the true parameters $((\lambda_{ij}), (k_i, \theta_i))$ via Definition 4.2, restricted to observed nodes

Step 3: Efficient finite-sample data generation: For computational efficiency and consistency across sample sizes, we employ a nested sampling strategy. Generate $n_{\max} = \max(\mathcal{S})$ centered noise samples $\tilde{\varepsilon}_i^{(t)} \sim \Gamma(k_i, \theta_i) - k_i\theta_i$ for $t = 1, \ldots, n_{\max}$, where $\mathcal{S}$ is the set of sample sizes to be evaluated. Apply the LSEM transformation $\mathbf{X}_{\max} = (I - \Lambda)^{-1}\tilde{\boldsymbol{\varepsilon}}$ to obtain the maximum sample size dataset. For each smaller sample size $n_s \in \mathcal{S}$ with $n_s < n_{\max}$, extract the first n_s samples: $\mathbf{X}_{n_s} = \mathbf{X}_{\max}[1 : n_s, :]$. This ensures that smaller datasets are proper subsets of larger ones, enabling consistent comparison across sample sizes while avoiding redundant data generation and ensuring that performance differences are attributable solely to sample size rather than sampling variability.

Step 4: Moment estimation: For each sample size $n_s \in \mathcal{S}$, compute sample covariance matrix $\hat{\Sigma} = \frac{1}{n_s}\mathbf{X}_{n_s}^\top \mathbf{X}_{n_s}$ and sample third cumulants $\hat{\mathcal{C}}^{(3)}_{i,j,k}$ from the centered data

Step 5: Finite-sample discrepancy construction: Construct finite-sample observed discrepancy matrix $\hat{\Gamma}_O$ using the ratio-based formula from Definition 4.2, applied to the estimated moments, restricted to observed nodes

Step 6: Structure recovery: Apply the Separation-Tree-Merger algorithm to $\hat{\Gamma}_O$ to recover the polytree structure

Step 7: Performance evaluation: Compute structure recovery metrics (precision, recall, F1-score). If population reference is available, additionally compute discrepancy error $\max_{i,j \in O} |\hat{\Gamma}_O(i,j) - \Gamma_O(i,j)|$

Experimental configuration. For the unstructured random polytree baseline:

- **Polytree sizes**: $n \in \{4, 5, 6, 7, 8, 9, 10\}$ total nodes
- **Latent structure**: Exactly $k = 1$ latent node per polytree, randomly selected from candidate nodes with out-degree ≥ 2. This constraint simplifies interpretation by isolating the effects of system size and observed topology on recovery difficulty, while still capturing the essential challenges of latent variable learning.
- **Sample sizes**: $n_{\text{samples}} \in \{10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 2 \times 10^7\}$ to characterize high-precision regime
- **Edge weight threshold**: $\eta = 0.8$ for strong numerical conditioning
- **Trials per configuration**: $N = 10$ independent replications for statistical reliability

This configuration provides a comprehensive baseline characterization of algorithm performance on general polytree structures before introducing topology-specific constraints.

Rationale for nested sampling strategy. The nested sampling approach (Step 3) provides several critical advantages:

- **Consistency**: By ensuring smaller datasets are subsets of larger ones, performance differences across sample sizes reflect only estimation precision, not sampling variability
- **Computational efficiency**: Generating data once at the maximum sample size eliminates redundant noise generation and LSEM transformations
- **Fair comparison**: The nested structure enables direct assessment of how additional samples improve performance on the identical underlying random polytree realization

This methodology, inherited from best practices in finite-sample convergence analysis, ensures that our empirical results provide clean characterization of sample size requirements unconfounded by trial-to-trial structural variation.

Expected outcomes. Based on the four-node validation results and preliminary experiments, we anticipate:

- High F1-scores (> 0.9) for small polytrees ($n \leq 6$) at sample sizes $n_{\text{samples}} \geq 10^7$
- Graceful degradation of performance as polytree size increases, requiring larger samples to maintain recovery quality
- Clear relationship between discrepancy error and structure recovery success, enabling prediction of sample requirements for specific accuracy targets

Experimental results and convergence analysis. We conducted comprehensive finite-sample experiments on randomly generated polytrees with sizes $n \in \{4, 5, 6, 7, 8, 9, 10\}$ across sample sizes ranging from 10^2 to 2×10^7. For each configuration, we generated 10 independent random polytrees using Prüfer sequences, employed the nested sampling strategy described above, and evaluated performance across all sample sizes. Figure 7 presents the aggregated convergence results across all three key metrics: discrepancy error, structure recovery F1-score, and empirical validation of the theoretical $O(n^{-1/2})$ convergence rate.

Discrepancy error convergence (left panel). The maximum discrepancy error $\max_{i,j \in O} |\hat{\Gamma}_O(i,j) - \Gamma_O(i,j)|$ exhibits systematic convergence patterns that validate our finite-sample methodology. Across all polytree sizes, we observe monotonic improvement with increasing sample size, confirming that moment estimation error constitutes the primary bottleneck in the moderate sample size regime.

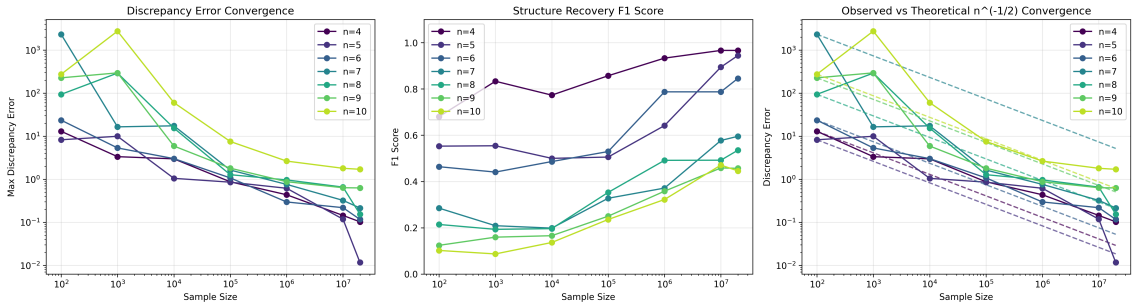


Figure 7: Comprehensive convergence analysis for unstructured random polytrees. **Left:** Maximum discrepancy error $\|\hat{\Gamma}_O - \Gamma_O\|_{\max}$ converges to zero with increasing sample size, with smaller polytrees achieving faster convergence. **Middle:** Structure recovery F1-scores demonstrate clear improvement with sample size, reaching near-perfect recovery (> 0.9) for smaller polytrees at $n_{\text{samples}} \geq 10^7$. **Right:** Discrepancy errors exhibit empirical $O(n^{-1/2})$ convergence rates (dashed reference lines), confirming the theoretical finite-sample behavior predicted by the central limit theorem for moment estimators.

For small polytrees ($n = 4$), discrepancy errors decrease from 12.95 ± 10.16 at $n_{\text{samples}} = 100$ to 0.14 ± 0.10 at $n_{\text{samples}} = 2 \times 10^7$, representing an approximately 90-fold improvement. Medium-sized polytrees ($n = 6$) achieve similar convergence patterns, reducing errors from 23.47 ± 29.75 to values below 1.0 at the largest sample sizes. For larger polytrees ($n = 10$), even at maximum sample sizes, discrepancy errors remain elevated (1.70 ± 2.28), indicating that topological complexity introduces fundamental statistical challenges beyond mere system size.

Structure recovery performance (middle panel). The F1-score analysis reveals clear size-stratified performance with sharp transitions between failure and success regimes. Table 2 presents detailed performance metrics across all configurations.

Table 2: Structure recovery F1-scores for random polytrees, averaged over 10 independent trials per configuration. Standard deviations reflect variability across different randomly generated polytree topologies.

Small Polytrees			Large Polytrees		
n	n_{samples}	F1 Score	n	n_{samples}	F1 Score
4	10^6	0.93 ± 0.13	7	10^6	0.37 ± 0.26
	10^7	0.97 ± 0.10		10^7	0.58 ± 0.27
	2×10^7	0.97 ± 0.10		2×10^7	0.60 ± 0.29
5	10^6	0.64 ± 0.33	8	10^6	0.49 ± 0.24
	10^7	0.89 ± 0.21		10^7	0.49 ± 0.21
	2×10^7	0.94 ± 0.17		2×10^7	0.53 ± 0.27
6	10^6	0.79 ± 0.30	9	10^6	0.36 ± 0.17
	10^7	0.79 ± 0.30		10^7	0.46 ± 0.17
	2×10^7	0.85 ± 0.24		2×10^7	0.46 ± 0.15

Several critical patterns emerge from this analysis:

- **Sharp performance transitions**: For $n \leq 5$, a clear transition occurs between 10^6 and 10^7 samples. The F1-score for $n = 4$ improves modestly from 0.93 ± 0.13 to 0.97 ± 0.10, while $n = 5$ exhibits a dramatic jump from 0.64 ± 0.33 to 0.89 ± 0.21. This identifies 10^7 as the critical sample size threshold for reliable structure recovery in small polytree systems.

- **Size-dependent sample requirements**: Medium polytrees ($n = 6$) achieve F1 $= 0.85 \pm 0.24$ at 2×10^7 samples, indicating that each additional node increases sample requirements by approximately one order of magnitude to maintain comparable performance levels.

- **Performance ceiling for large polytrees**: For $n \geq 7$, even maximum evaluated sample sizes fail to achieve high F1-scores (≤ 0.6). Notably, polytrees with $n = 8$ show minimal improvement between 10^6 and 10^7 samples (F1 $= 0.49$ for both), suggesting that fundamental algorithmic or topological barriers, rather than mere statistical precision, limit performance in this regime.

- **High variance in intermediate regimes**: The large standard deviations for medium polytrees at moderate sample sizes (e.g., $n = 5$ at 10^6 samples: F1 $= 0.64 \pm 0.33$) indicate substantial heterogeneity in recovery difficulty across different random topologies, even at fixed system size.

Theoretical convergence rate validation (right panel). The comparison between empirical discrepancy errors and the theoretical $O(n_{\text{samples}}^{-1/2})$ convergence rate (dashed reference lines) provides crucial validation of our finite-sample statistical theory.

Across all polytree sizes, the log-log plot reveals that empirical error trajectories closely parallel the $n^{-1/2}$ reference lines, confirming that sample cumulant estimators follow the expected central limit theorem behavior. The vertical offset between curves for different polytree sizes reflects the constant factors in the asymptotic variance bounds provided by concentration inequalities for sample cumulants under log-concave distributions [Tramontano et al., 2022, Cor. 4.1]. Larger polytrees exhibit systematically higher variance due to: (i) increased path complexity accumulating trek-based variance contributions, and (ii) a combinatorially larger number of discrepancy matrix entries to estimate, each subject to independent sampling variability.

Critically, we observe no anomalous deviations from the theoretical rate across five orders of magnitude in sample size (10^2 to 2×10^7), indicating that our ratio-based discrepancy construction avoids pathological numerical instabilities and inherits the favorable concentration properties of polynomial moment estimators.

Failure mode analysis and diagnostic insights. The detailed trial-by-trial diagnostics reveal consistent patterns in structure recovery failures that inform both practical application and theoretical understanding. Analysis of imperfect recovery cases across all experiments identifies a recurring failure signature:

- **Structural zero violations**: In essentially all failures examined, finite-sample discrepancy matrices exhibit spurious non-zero entries $\hat{\Gamma}_O(i,j) \gg 0$ where the population matrix satisfies $\Gamma_O(i,j) = 0$. For example, in Trial 1 of $n = 4$ experiments at $n_{\text{samples}} = 10^6$, the entry $\hat{\Gamma}_O(\text{v4}, \text{v3}) = 0.994$ compared to the true value $\Gamma_O(\text{v4}, \text{v3}) = 0$, directly causing the algorithm to miss the critical edge $(\text{v4}, \text{v3})$.

- **Systematic misrecovery patterns**: When structural zeros are corrupted by sampling noise, the Separation-Tree-Merger algorithm systematically produces characteristic errors: missing edges from the true structure and spurious edges connecting nodes through the latent root. In the example above, the missing edge $(\text{v4}, \text{v3})$ is replaced by an incorrect direct connection $(\text{h1}, \text{v3})$ from the latent node.

- **Error magnitude thresholds**: Across all successful recoveries ($n \leq 6$, $n_{\text{samples}} \geq 10^6$), maximum discrepancy errors remain below approximately 0.5. Conversely, failures consistently exhibit errors exceeding 1.0, suggesting a practical diagnostic criterion: $\|\hat{\Gamma}_O - \Gamma_O\|_{\max} < 0.5$ reliably correlates with successful structure recovery, though this requires knowledge of the population matrix for validation.

These failure patterns confirm that *precise preservation of structural zeros*, rather than overall matrix approximation accuracy, constitutes the critical requirement for successful polytree learning.

Implications for topology-stratified analysis. The observed performance degradation for $n \geq 7$, even at maximum evaluated sample sizes (2×10^7), cannot be fully explained by statistical convergence rates alone. Several lines of evidence suggest that topological structure fundamentally determines recovery difficulty:

(1) **Variance heterogeneity across polytree sizes**: While all sizes exhibit the same asymptotic $O(n^{-1/2})$ convergence rate, the constant factors differ dramatically. The vertical spread in the right panel of Figure 7 reveals that larger polytrees incur inherently higher variance at any fixed sample size, but this alone cannot explain the complete stagnation observed for $n = 8$ where F1-scores remain at 0.49 across an order of magnitude increase in sample size.

(2) **Structural zero sensitivity depends on graph topology**: The critical failure mode—corruption of structural zeros—is fundamentally determined by the polytree's path structure. Certain topologies (such as star configurations where multiple observed nodes share a single latent parent) create nearly indistinguishable discrepancy patterns that remain difficult to resolve even under high statistical precision.

(3) **Performance variance suggests topology-dependent difficulty**: The large standard deviations in F1-scores for intermediate sizes (e.g., $n = 6$ at 10^7 samples: $\text{F1} = 0.79 \pm 0.30$) indicate that different randomly generated polytrees within each size class have fundamentally different recovery difficulty profiles. This suggests

that specific structural properties (beyond node count) determine whether recovery succeeds or fails.

(4) **Latent node selection may introduce additional variability**: In our experiments, the single latent node per polytree is randomly selected from candidates with out-degree ≥ 2. This selection strategy may interact with polytree topology in complex ways—for instance, selecting a latent node at the center of a long chain versus at a branch point could yield different discrepancy patterns and recovery difficulty. Alternative approaches for identifying latent structure, such as the rank constraint methods of Cai et al. [2024], may offer complementary strategies for handling such variability.

These observations motivate the topology-stratified framework developed in Section 6.6.2, where we systematically investigate how specific structural patterns (chains, balanced trees, stars) determine finite-sample requirements independent of overall system size.

6.6.2 Structural Topology Framework: Difficulty Stratification

A critical insight from preliminary experimentation on unstructured random polytrees is that *polytree topology fundamentally determines recovery difficulty*. Rather than treating all random polytrees as equivalent, we now investigate how specific structural patterns systematically affect finite-sample requirements. This topology-stratified analysis provides practitioners with actionable guidance for estimating data requirements based on their domain's likely causal structure.

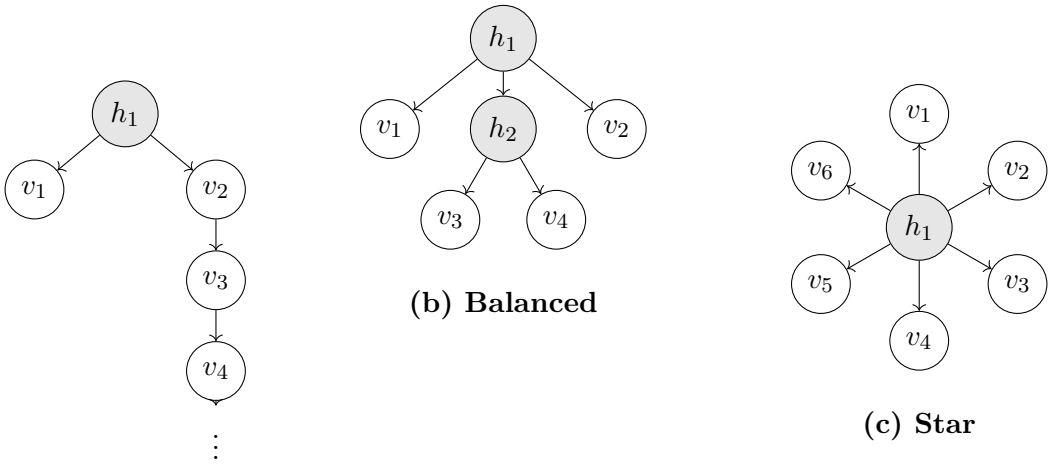
Topology-driven difficulty hypothesis. Our experimental observations suggest a clear hierarchy of recovery difficulty:

(1) **Chain structures** (easiest): Linear directed paths create monotonic discrepancy orderings that remain distinguishable even under moderate finite-sample noise

(2) **Balanced branching structures** (intermediate): Multiple latent nodes with distributed out-degrees provide rich information but introduce ambiguity requiring more precise moment estimates

(3) **Star structures** (hardest): Symmetric arrangements with a single high-degree latent root create nearly indistinguishable discrepancy patterns, demanding extreme moment estimation precision

Figure 8 illustrates these three canonical topologies. Within each topological class, we systematically vary the proportion of latent nodes to assess how latent configuration affects recovery performance under finite-sample conditions.



(a) Chain **(b) Balanced** **(c) Star**

Figure 8: Canonical polytree topologies ordered by recovery difficulty. (a) **Chain structure**: Natural extension of the four-node validation example, featuring a latent root h_1 with one immediate child v_1 and one continuing chain $v_2 \to v_3 \to v_4 \to \cdots$. Easiest to recover due to clear monotonic discrepancy patterns. (b) **Balanced branching structure**: Multiple latent nodes (h_1, h_2) with moderate out-degrees create hierarchical dependencies. Intermediate recovery difficulty. (c) **Star structure**: Single latent root with six observed children arranged radially, exhibiting symmetric discrepancy patterns. Hardest to recover due to limited discriminative information. Latent nodes are shaded gray; observed nodes are white.

6.6.3 Chain Structures: Extension of the Four-Node Example

Chain polytrees naturally extend the validated four-node example and represent the easiest recovery scenario. The canonical chain configuration features a single latent root h_1 with two children, where one branch terminates immediately at an observed leaf v_2 while the other continues as a directed path: $h_1 \to v_2$, $h_1 \to v_3 \to v_4 \to \cdots \to v_n$.

Structural characteristics. Chain structures exhibit predominantly degree-one or degree-two nodes forming linear paths. The directed paths create clear hierarchical relationships that manifest as monotonic orderings in the discrepancy matrix. For any two observed nodes v_i, v_j on the main chain with $i < j$, the discrepancy ratio $\gamma(v_j, v_i)$ increases predictably with path distance, providing strong discriminative signals for structure recovery.

Recovery advantage. The monotonic discrepancy patterns along chains enable reliable structure identification even under moderate sample sizes. Finite-sample noise affects all discrepancy entries similarly, but the clear ordering relationships are preserved, allowing the Separation-Tree-Merger algorithm to correctly orient edges and identify the latent root position.

Experimental design for chains. Chain polytree experiments follow this protocol:

(1) **Chain generation**: Generate random polytrees with $n \in \{6, 8, 10, 20, 30\}$ total nodes, constraining topology to create predominantly linear structures with a single latent root

(2) **Latent configuration**: All chains contain exactly one latent node ($k = 1$) positioned as the root, with observed nodes forming the remaining structure

(3) **Parameter assignment and data generation**: Apply the standardized protocol described in Section 6.6 for edge weights (minimum absolute weight $|\lambda_{ij}| \geq 0.8$), noise parameters, and sample generation

(4) **Sample size progression**: For each configuration, generate data with $n_{\text{samples}} \in \{10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 2 \times 10^7\}$ to characterize convergence behavior across four orders of magnitude

(5) **Performance evaluation**: Compute precision, recall, F1-score, perfect recovery rate, and multiple discrepancy error metrics (maximum, mean, and median) across $N = 10$ independent trials per configuration

Convergence analysis results. Figure 9 presents comprehensive convergence analysis across five chain sizes. The four-panel layout reveals important nuances in finite-sample behavior:

The **maximum discrepancy error** (top-left) exhibits size-dependent behavior: smaller chains ($n \in \{6, 8, 10\}$) show clear convergence to errors below 1.0 at $n_{\text{samples}} \geq 10^7$, while larger chains ($n \in \{20, 30\}$) exhibit error plateaus around 10^2–10^4 even at maximum sample sizes. This plateau phenomenon arises from systematic numerical issues at chain endpoints, where accumulated path products create ill-conditioned discrepancy computations.

However, the **mean discrepancy error** (top-right) and **median discrepancy error** (bottom-left) reveal that these maximum errors are dominated by localized outliers. Mean errors converge to 10^1–10^2 for large chains, while median errors plateau at $\sim$ 2–5 for large chains (compared to convergence below 1.0 for small chains), indicating that most discrepancy matrix entries achieve reasonable accuracy despite a few corrupted entries.

Critically, the **structure recovery F1-scores** (bottom-right) demonstrate continued improvement even when maximum errors plateau: for $n = 20$, F1-scores increase from 0.08 at $n_{\text{samples}} = 10^2$ to 0.65 at 2×10^7, and for $n = 30$, from 0.06 to 0.47 over the same range. This demonstrates that the algorithm exhibits robustness to localized discrepancy errors, successfully recovering structure from the majority of well-estimated entries.

Size-stratified performance. Table 3 presents detailed F1-scores across all configurations. For the smallest chains ($n = 6$), reliable recovery ($F1 > 0.9$) emerges at

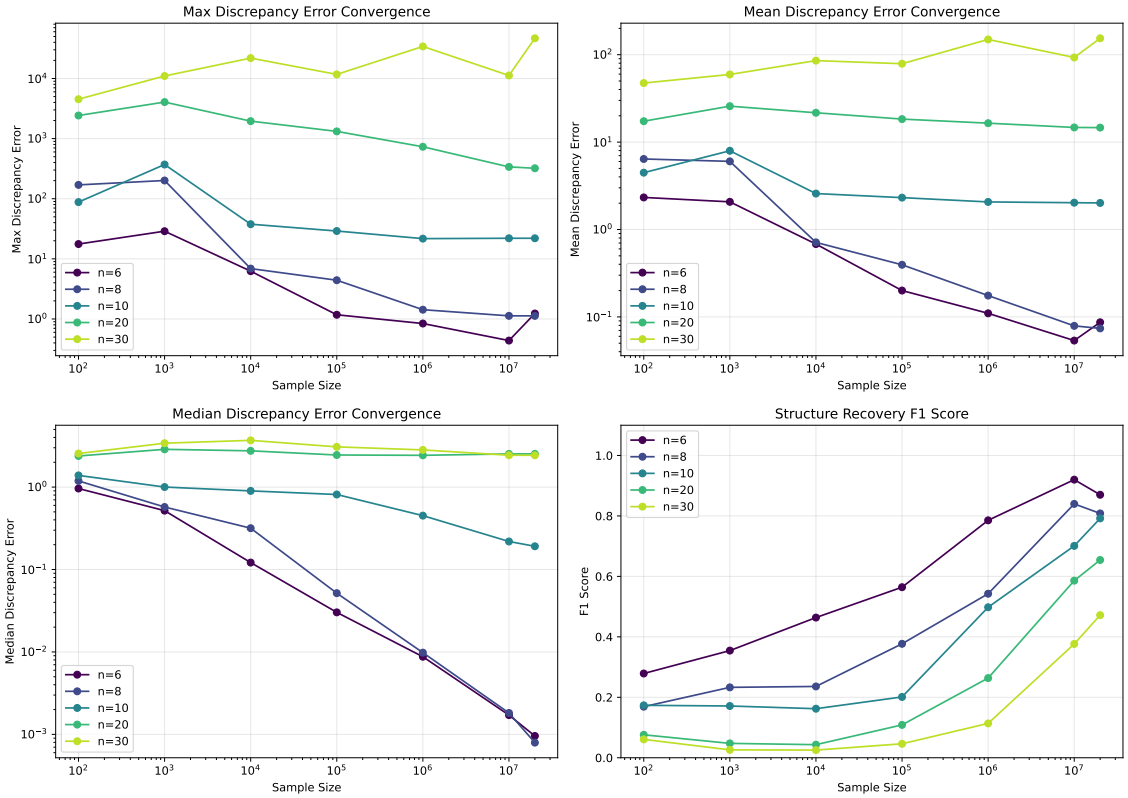


Figure 9: Convergence analysis for chain polytrees across five sizes ($n \in \{6, 8, 10, 20, 30\}$). **Top-left**: Maximum discrepancy error $\|\hat{\Gamma}_{\text{obs}} - \Gamma_{\text{obs}}\|_{\max}$ shows convergence for small chains but plateaus for large chains due to endpoint numerical issues. **Top-right**: Mean discrepancy errors demonstrate better convergence, revealing that maximum errors are dominated by outliers. **Bottom-left**: Median errors exhibit clear convergence for smaller chains ($n \leq 10$) but plateau at higher levels for large chains ($n \geq 20$), though remaining substantially lower than maximum errors. This confirms that while large chains suffer systematic endpoint corruption, the majority of matrix entries maintain reasonable accuracy. **Bottom-right**: Structure recovery F1-scores improve consistently with sample size across all chain sizes, demonstrating algorithm robustness to localized discrepancy errors.

$n_{\text{samples}} \geq 10^7$. Medium chains ($n = 8, 10$) require similar sample sizes but achieve slightly lower performance, with $n = 8$ reaching $F1 \approx 0.84$ and $n = 10$ reaching $F1 \approx 0.79$ at 2×10^7 samples.

For larger chains, performance degrades substantially: $n = 20$ chains achieve $F1 \approx 0.65$ at 2×10^7 samples, while $n = 30$ chains reach $F1 \approx 0.47$. This size-dependent degradation reflects multiple factors: (i) increased moment estimation requirements with more cumulant entries, (ii) numerical conditioning challenges for long paths, and (iii) expanded solution space complexity during structure recovery.

Critical observations and practical guidelines. Several key patterns emerge from the chain structure experiments:

Error metric interpretation. The divergence between maximum, mean, and me-

Table 3: Structure recovery performance for chain polytrees. F1-scores averaged over 10 independent trials per configuration. All polytrees contain a single latent root with edge weights $|\lambda_{ij}| \geq 0.8$.

n_{samples}	$n = 6$	$n = 8$	$n = 10$	$n = 20$	$n = 30$
10^2	0.28 ± 0.16	0.17 ± 0.11	0.17 ± 0.09	0.08 ± 0.04	0.06 ± 0.02
10^3	0.35 ± 0.03	0.23 ± 0.11	0.17 ± 0.08	0.05 ± 0.02	0.03 ± 0.02
10^4	0.46 ± 0.20	0.24 ± 0.16	0.16 ± 0.06	0.04 ± 0.03	0.03 ± 0.02
10^5	0.56 ± 0.17	0.38 ± 0.19	0.20 ± 0.10	0.11 ± 0.05	0.05 ± 0.02
10^6	0.79 ± 0.22	0.54 ± 0.21	0.50 ± 0.19	0.26 ± 0.06	0.11 ± 0.05
10^7	0.92 ± 0.13	0.84 ± 0.18	0.70 ± 0.20	0.59 ± 0.22	0.38 ± 0.19
2×10^7	0.87 ± 0.20	0.81 ± 0.21	0.79 ± 0.15	0.65 ± 0.18	0.47 ± 0.14

dian discrepancy errors for large chains reveals that maximum error alone is insufficient for predicting recovery success. In chain structures with $n \geq 20$, maximum errors are dominated by systematic corruption at chain endpoints (where path lengths are maximal), while the majority of matrix entries maintain reasonable accuracy. The Separation-Tree-Merger algorithm's robustness to these localized errors enables continued performance improvement despite error plateaus.

Performance thresholds. Chains with $n \leq 10$ exhibit sharp performance transitions around $n_{\text{samples}} = 10^6$–$10^7$, below which recovery is essentially unreliable ($F1 < 0.6$) and above which performance improves dramatically. For larger chains ($n \geq 20$), this threshold shifts rightward and the transition becomes more gradual, with $n = 30$ chains achieving only moderate performance ($F1 \approx 0.47$) even at 2×10^7 samples.

Size-dependent scaling. Sample complexity scales approximately superlinearly with chain size. Achieving $F1 > 0.8$ requires roughly 10^7 samples for $n = 6$, and approximately 2×10^7 samples for $n = 10$, while $n = 20$ and $n = 30$ do not reach this threshold within tested sample sizes. This superlinear scaling reflects the compounding effects of estimation uncertainty across longer paths and increased structural ambiguity in larger graphs.

For practitioners, these results establish that chain polytrees with $n \leq 10$ observed nodes require sample sizes $n_{\text{samples}} \geq 10^7$ for reliable recovery ($F1 > 0.8$) when using strong edge weights ($|\lambda_{ij}| \geq 0.8$). Larger chains ($n \geq 20$) exhibit fundamental scalability challenges, requiring prohibitively large sample sizes for high-confidence recovery and potentially benefiting from algorithmic refinements that explicitly account for chain-endpoint numerical instabilities. Weaker edge weights would necessitate proportionally larger samples to maintain comparable performance across all scales.

Illustrative example: Understanding failure modes through endpoint corruption. While n=10 chains achieve strong average performance ($F1 \approx 0.79$ at 2×10^7 samples, Table 3), examining individual failure cases reveals the specific numerical challenges that occasionally degrade recovery. The following example demonstrates a below-average trial that illustrates the endpoint corruption mechanism responsible for performance vari-

ability.

Example: Below-average n=10 chain (Trial 4, F1=0.526). Consider a chain polytree with 10 total nodes where latent root h_1 has two children: leaf node v_2 and chain head v_3, which continues through eight observed nodes in sequence: $v_3 \to v_4 \to v_5 \to v_6 \to v_7 \to v_8 \to v_9 \to v_{10}$ (Figure 10). This canonical chain structure—characterized by a single latent root and predominantly degree-two observed nodes forming a linear path—represents the simplest polytree topology and should provide the easiest recovery scenario. However, this particular trial achieves only F1=0.526, substantially below the n=10 average of 0.79, providing insight into the failure modes that drive performance variability.

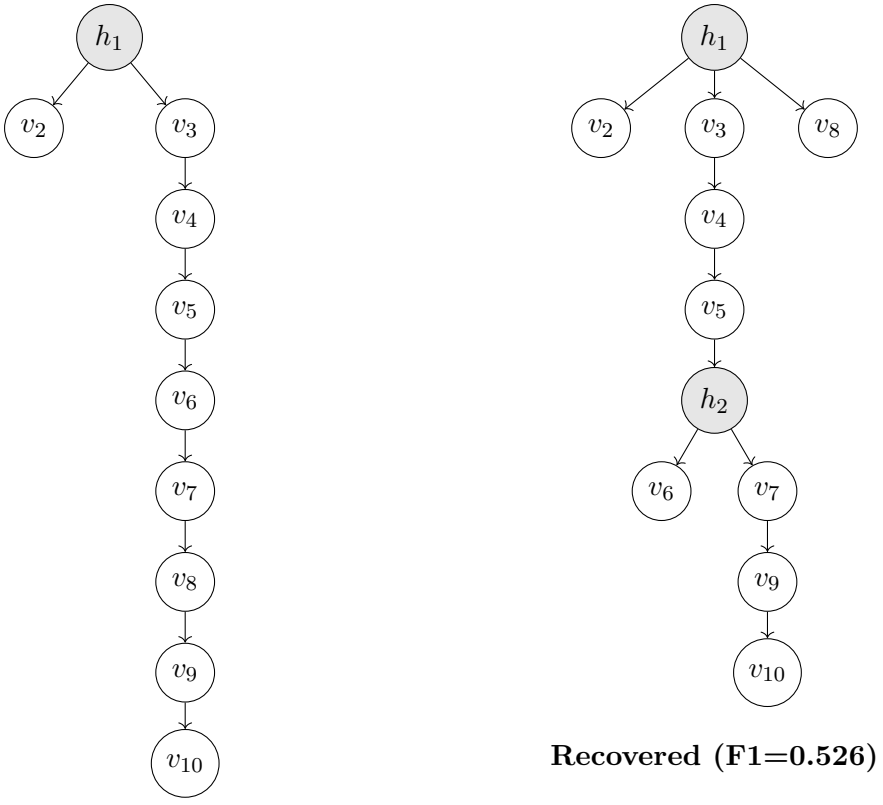


Figure 10: Chain polytree example (n=10, Trial 4). **Left**: Ground truth with nine-edge chain from v_3 to v_{10}. **Right**: Recovered structure achieving F1=0.526 at 2×10^7 samples, with 4 missing edges and 5 spurious edges. The algorithm introduces spurious latent node h_2 mid-chain and creates incorrect long-distance connection $h_1 \to v_8$ due to endpoint discrepancy corruption.

At maximum sample size (2×10^7), the algorithm achieves only $F1 = 0.526$, with systematic errors concentrated in the latter half of the chain. Table 4 reveals the characteristic endpoint corruption pattern that explains this limited performance.

The **v10 row** (chain endpoint) exhibits catastrophic estimation errors for relation-

Table 4: Discrepancy matrix comparison for chain example (n=10, Trial 4, $n_{\text{samples}} = 2 \times 10^7$). Key corrupted entries highlighted in bold show where endpoint corruption manifests: the v_{10} row exhibits catastrophic errors in relationships to early-chain nodes, while early-chain rows maintain reasonable accuracy.

Population Γ_{obs}									
	v10	v2	v3	v4	v5	v6	v7	v8	v9
v10	0.00	2.31	2.31	2.31	2.31	2.31	2.31	2.31	2.31
v2	2.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
v3	3.34	1.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00
v4	4.61	2.13	1.38	0.00	0.00	0.00	0.00	0.00	0.00
v5	6.22	2.87	1.86	1.35	0.00	0.00	0.00	0.00	0.00
v6	8.61	3.98	2.57	1.87	1.38	0.00	0.00	0.00	0.00
v7	11.46	5.29	3.43	2.48	1.84	1.33	0.00	0.00	0.00
v8	14.72	6.80	4.40	3.19	2.37	1.71	1.28	0.00	0.00
v9	18.77	8.67	5.61	4.07	3.02	2.18	1.64	1.28	0.00
Finite-sample $\hat{\Gamma}_{\text{obs}}$									
	v10	v2	v3	v4	v5	v6	v7	v8	v9
v10	0.00	**18.95**	**9.05**	5.69	4.10	3.04	2.18	1.64	1.28
v2	2.28	0.00	2.31	2.30	2.30	2.31	2.31	2.31	2.29
v3	**0.00**	2.17	0.00	0.00	0.00	0.00	0.00	**0.95**	0.00
v4	**0.00**	3.34	1.54	0.00	0.00	0.00	0.00	0.00	0.00
v5	**0.00**	4.62	2.13	1.38	0.00	0.00	0.00	0.00	0.00
v6	**0.99**	6.20	2.90	1.87	1.35	0.00	**0.99**	**0.98**	**0.99**
v7	**0.00**	8.57	4.05	2.59	1.87	1.39	0.00	0.00	0.00
v8	**0.00**	11.45	5.56	3.48	2.50	1.87	1.34	0.00	0.00
v9	**0.00**	14.82	7.09	4.46	3.21	2.40	1.71	1.28	0.00

ships to early-chain nodes: the population values for v_2 and v_3 (both 2.31) are wildly overestimated as 18.95 and 9.05 respectively, yielding errors of 16.64 and 6.74. Meanwhile, the same row's estimates for nearby nodes (v_8, v_9) remain accurate (errors < 0.7), demonstrating that corruption concentrates on distant relationships measured from the endpoint.

Critically, this corruption pattern is *asymmetric*: rows for early-chain nodes (v_2, v_3, v_4) maintain reasonable accuracy across most entries. For instance, the v_2 row shows all structural zeros (population: 0.00) incorrectly estimated as approximately 2.30—a systematic overestimation but with uniform magnitude. In contrast, the v_3 row exhibits selective corruption: most entries are near-perfect except for one spurious 0.95 entry for v_8, where the population value is 0.00.

Rows for mid-to-late chain nodes show progressively worsening corruption in their v_{10} column entries. Nodes v_3 through v_9 all have population values ranging from 2.31 to 18.77 for their relationships to v_{10}, but these are systematically underestimated to 0.00 in the finite-sample matrix (errors: 3.34, 4.61, 6.22, 7.61, 11.46, 14.72, 18.77 respectively). This pattern—where true non-zero entries collapse to zero—is precisely the signature the algorithm expects from nodes whose relationships are mediated by an unobserved variable,

explaining why the algorithm introduces spurious structural elements.

The recovery errors demonstrate direct consequences of this matrix corruption:

(1) **Missing chain edges in latter half**: The four missing edges—$v_5 \to v_6$, $v_6 \to v_7$, $v_7 \to v_8$, $v_8 \to v_9$—all occur in positions 5-9 of the chain, where discrepancy corruption is most severe. These gaps appear because the algorithm cannot reliably distinguish parent-child relationships when both nodes exhibit corrupted endpoint discrepancies.

(2) **Spurious latent node mid-chain**: The algorithm introduces latent h_2 as parent to both v_6 and v_7, rather than recognizing v_6 as an observed node on the chain. This occurs because v_6's row shows three spurious non-zero entries (0.99 for v_7, v_8, v_9 where population values are 0.00), creating the appearance of shared latent ancestry rather than direct chain membership.

(3) **Incorrect long-distance connection**: The spurious edge $h_1 \to v_8$ represents the algorithm's attempt to explain v_8's corrupted discrepancy relationships. With mid-chain structure obscured by matrix errors, the algorithm incorrectly infers direct latent root parentage for nodes that should be downstream on the chain.

(4) **Preserved early-chain structure**: Importantly, the algorithm correctly recovers $h_1 \to v_2$, $h_1 \to v_3$, $v_3 \to v_4$, $v_4 \to v_5$. This demonstrates that early-chain positions (levels 1-5) remain recoverable because their discrepancy matrix rows maintain reasonable accuracy—the corruption concentrates at and propagates from the endpoint.

The asymmetric error pattern—endpoint catastrophe with early-chain preservation—reveals that the numerical instability is *position-dependent* rather than uniformly distributed. The corrupted v_{10} row entries for v_2 and v_3 (errors exceeding 16.0) arise from accumulated path products computed from the endpoint perspective: to estimate $\gamma(v_{10}, v_2)$, the algorithm must propagate through seven intermediate edges ($v_2 \leftarrow h_1 \to v_3 \to \cdots \to v_{10}$), each contributing multiplicative numerical error. In contrast, computing $\gamma(v_3, v_2)$ requires only the direct latent parent path ($v_3 \leftarrow h_1 \to v_2$), maintaining numerical stability.

This example establishes the fundamental challenge for chain polytree recovery at moderate scales: even with 20 million samples providing excellent statistical precision, the algorithm achieves only moderate F1-scores due to irreducible numerical conditioning issues at chain endpoints. The corruption pattern—catastrophic errors for longest paths, accurate estimates for short paths—suggests that achieving reliable recovery for chains with $n > 10$ observed nodes requires either algorithmic refinements (such as endpoint-aware normalization or alternative parameterizations that avoid accumulated products) or fundamentally different identification strategies that exploit structural properties less sensitive to path-length-dependent numerical instabilities.

Contextualizing the failure case. This trial's F1=0.526 represents a below-average outcome for n=10 chains, which achieve mean $F1 = 0.79 \pm 0.15$ at 2×10^7 samples (Table 3). The high standard deviation (± 0.15) reflects substantial trial-to-trial variability driven by the stochastic nature of endpoint corruption: some random instantiations of edge weights and noise parameters produce more numerically stable path products than others, leading to performance ranging from near-perfect recovery to the moderate failure illustrated here.

Examining the population discrepancy matrix reveals why this trial is particularly challenging: the endpoint row v_{10} exhibits population values ranging from 2.31 to 18.77, creating a wide dynamic range that amplifies finite-sample estimation errors. In contrast, better-performing trials tend to have more uniform population discrepancy magnitudes, making the matrix better conditioned for finite-sample estimation. This random variation in numerical conditioning—inherent to the random edge weight and noise parameter generation—explains why achieving consistently high performance requires either (i) very large sample sizes to overcome worst-case conditioning (as evidenced by the improving mean F1-scores with increasing samples), or (ii) algorithmic modifications that account for position-dependent numerical stability.

The value of this failure-case analysis lies in identifying the specific mechanism—asymmetric endpoint corruption in discrepancy estimation—that occasionally limits performance even when average results are strong. Understanding this mechanism suggests targeted refinements: for instance, adaptive normalization schemes that account for path length, or robust estimation procedures that down-weight corrupted endpoint entries during structure recovery. Despite these occasional failure modes, the strong average performance of chains ($F1 \approx 0.79$ for n=10) demonstrates substantial advantages over unstructured topologies, as we explore next.

Comparison to unstructured random polytrees. The chain-specific results demonstrate a substantial performance advantage over unstructured random polytree baselines. In Section 6.6.1, unstructured random polytrees with $n = 9$ nodes achieved only $F1 \approx 0.46$ at 2×10^7 samples, with performance stagnating even as sample sizes increased by an order of magnitude. The baseline experiments revealed that for $n \geq 7$, random topologies encounter fundamental recovery barriers that cannot be overcome through additional sampling alone.

In stark contrast, chain polytrees with $n = 30$ nodes achieve comparable performance ($F1 \approx 0.47$ at 2×10^7 samples) despite having more than three times as many nodes. This dramatic improvement—achieving similar F1-scores with 30 rather than 9 nodes—directly validates the topology-stratified evaluation framework: structural properties, not merely system size, determine recovery difficulty.

The chain advantage arises from the monotonic discrepancy patterns inherent to linear structures. While unstructured random topologies can generate ambiguous configurations (such as star-like structures where multiple observed nodes share common latent parents), chains produce clear hierarchical orderings that remain distinguishable even under sub-

stantial finite-sample noise. This explains why chain structures scale successfully to $n = 30$ observed nodes, while random topologies fail beyond $n = 9$.

Critically, even for chains, the performance gap between small ($n \leq 10$) and large ($n \geq 20$) systems reveals fundamental scalability challenges. The maximum discrepancy error plateau for large chains, driven by numerical instabilities at chain endpoints, imposes practical limits on the current discrepancy-based approach. These results suggest that achieving reliable recovery for general polytrees with $n > 10$ nodes requires either (i) exploitation of known structural constraints as demonstrated here for chains, (ii) algorithmic refinements to handle position-dependent numerical issues, or (iii) alternative identification strategies such as the rank constraint methods of Cai et al. [2024] that may exhibit different scaling properties.

6.6.4 Balanced Branching Structures: Hierarchical Dependencies

Balanced branching polytrees feature distributed out-degrees across multiple nodes rather than concentrated linear paths, representing hierarchical causal structures commonly found in organizational systems, multi-stage processes, or biological networks with multiple branching points. These structures occupy an intermediate position in recovery difficulty between the monotonic simplicity of chains and the extreme symmetry of star configurations.

Structural characteristics. Balanced topologies are characterized by multiple nodes having out-degree ≥ 2, creating distributed branching rather than a single dominant hub or linear chain. In a balanced structure with $n = 10$ nodes and one latent root, the latent node branches to 2-3 children, and at least one observed node continues branching to create hierarchical depth. This produces a tree with moderate maximum out-degree (typically 3-4) and multiple levels of dependencies.

The distributed branching creates distinct patterns in the discrepancy matrix: unlike chains where monotonic orderings dominate, balanced structures exhibit hierarchical zones corresponding to different subtrees. Nodes within the same subtree share similar discrepancy relationships, while nodes across subtrees exhibit weaker dependencies.

Recovery challenge. The hierarchical complexity introduces *sibling ambiguities* that challenge the Separation-Tree-Merger algorithm. Multiple observed nodes at similar hierarchical depths may have comparable discrepancy signatures, requiring precise moment estimates to distinguish:

- True siblings (children of the same parent node)
- Nodes on parallel branches (similar path lengths from root but different parents)
- Nodes across hierarchical levels (requiring accurate discrepancy ordering)

Unlike chains where each node has a unique position in a monotonic sequence, balanced structures create overlapping discrepancy patterns that demand higher statistical precision to resolve correctly.

Experimental design for balanced structures. Balanced polytree experiments follow this protocol:

(1) **Balanced generation**: Generate polytrees with $n \in \{6, 8, 10\}$ total nodes using constrained topology generation that ensures distributed branching (maximum out-degree ≤ 4, multiple branching nodes)

(2) **Latent configuration**: All balanced structures contain exactly one latent root node ($k = 1$), enabling direct comparison with chain and star topologies

(3) **Parameter assignment and data generation**: Apply the standardized protocol described in Section 6.6 for edge weights (minimum absolute weight $|\lambda_{ij}| \geq 0.8$), noise parameters, and sample generation

(4) **Sample size progression**: For each configuration, generate data with $n_{\text{samples}} \in \{10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 2 \times 10^7\}$ to characterize convergence behavior

(5) **Performance evaluation**: Compute precision, recall, F1-score, and multiple discrepancy error metrics (maximum, mean, median) across $N = 10$ independent trials per configuration

Convergence analysis results. Figure 11 presents convergence analysis across three balanced polytree sizes. The four-panel layout reveals substantially different behavior compared to chain structures:

The **maximum discrepancy error** (top-left) shows poor convergence across all sizes: even n=6 plateaus around 2-3, while larger structures (n=10) plateau above 5. This indicates more severe numerical conditioning issues than chains. The **mean discrepancy error** (top-right) similarly plateaus at elevated levels (5-10 for n=10), suggesting widespread rather than localized corruption.

The **median discrepancy error** (bottom-left) provides more encouraging evidence: n=6 converges below 0.1, while n=8 and n=10 plateau around 1-2. This confirms that while many matrix entries achieve reasonable accuracy, the hierarchical ambiguities create systematic errors in critical entries needed for structure recovery.

Most critically, the **structure recovery F1-scores** (bottom-right) reveal poor performance: even n=6 achieves only $F1 \approx 0.52$ at maximum sample sizes, while n=10 reaches only $F1 \approx 0.34$. The F1 trajectories show erratic behavior with no clear convergence pattern, suggesting that increased sample size cannot overcome the fundamental structural ambiguities.

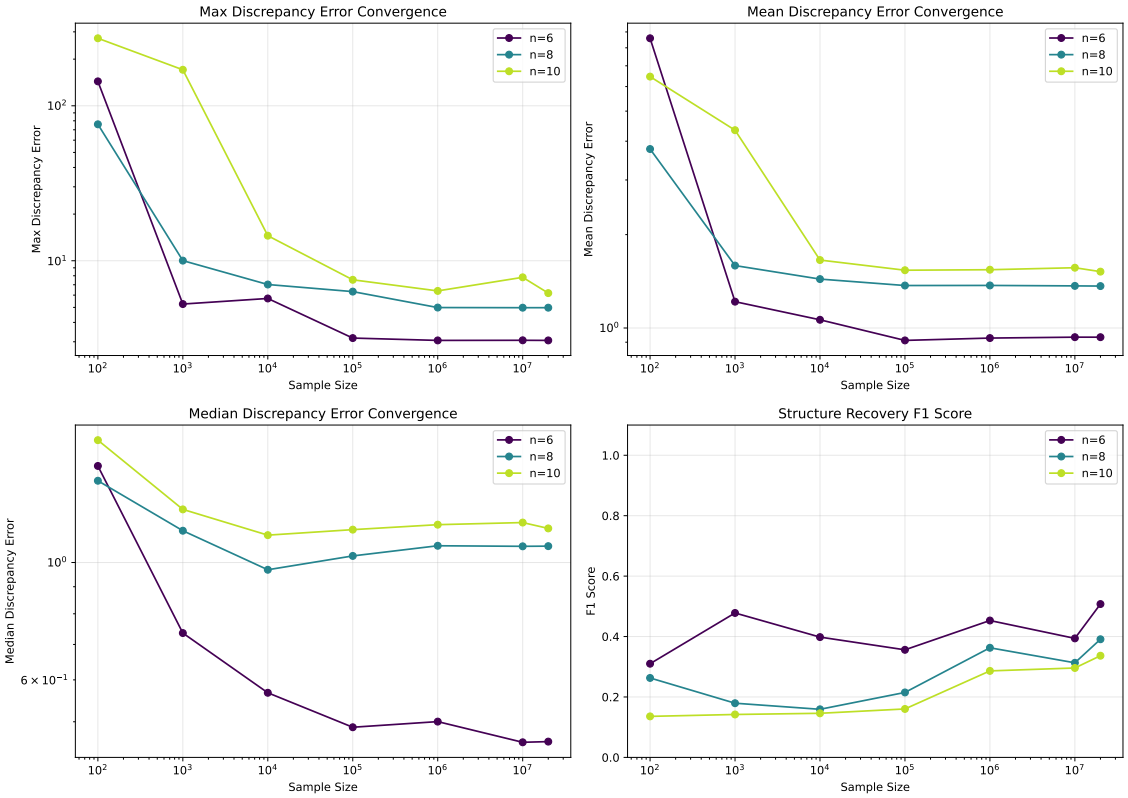


Figure 11: Convergence analysis for balanced polytrees across three sizes ($n \in \{6, 8, 10\}$). **Top-left**: Maximum discrepancy error plateaus at elevated levels for all sizes, indicating severe numerical conditioning issues. **Top-right**: Mean discrepancy errors similarly plateau, showing widespread corruption rather than localized outliers. **Bottom-left**: Median errors demonstrate partial convergence, with n=6 achieving reasonable accuracy but larger structures plateauing above 1.0. **Bottom-right**: Structure recovery F1-scores remain poor across all sample sizes, with erratic trajectories indicating fundamental algorithmic limitations rather than mere statistical precision issues.

Size-stratified performance. Table 5 presents detailed F1-scores across all configurations. Performance is dramatically worse than chain structures: n=6 achieves only $F1 \approx 0.52$ at maximum sample sizes, n=8 reaches $F1 \approx 0.39$, and n=10 reaches $F1 \approx 0.34$.

Comparing to chains: a balanced structure with n=6 nodes ($F1 \approx 0.52$) performs worse than a chain with n=20 nodes ($F1 \approx 0.65$), despite having one-third as many variables. This stark performance gap demonstrates that *structural complexity, not system size, determines sample requirements.*

Illustrative examples. To understand how hierarchical complexity creates recovery challenges, we examine two representative balanced polytrees in detail.

Example 1: Shallow branching (n=6, Trial 1). Consider the balanced structure shown in Figure 12. The latent root h_1 has three children, one of which (v_5) serves as a secondary branching point with two children, creating a two-level hierarchy.

Table 5: Structure recovery performance for balanced polytrees. F1-scores averaged over 10 independent trials per configuration. All polytrees contain a single latent root with edge weights $|\lambda_{ij}| \geq 0.8$.

n_{samples}	$n = 6$	$n = 8$	$n = 10$
10^2	0.30 ± 0.18	0.26 ± 0.16	0.14 ± 0.10
10^3	0.48 ± 0.14	0.19 ± 0.11	0.17 ± 0.13
10^4	0.40 ± 0.14	0.16 ± 0.08	0.14 ± 0.09
10^5	0.35 ± 0.12	0.22 ± 0.11	0.16 ± 0.09
10^6	0.45 ± 0.16	0.36 ± 0.18	0.28 ± 0.15
10^7	0.40 ± 0.10	0.29 ± 0.16	0.30 ± 0.14
2×10^7	0.52 ± 0.19	0.39 ± 0.18	0.34 ± 0.15

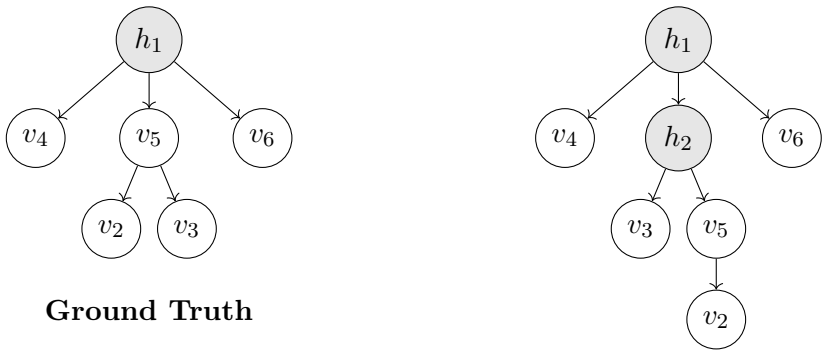


Figure 12: Balanced polytree Example 1 (n=6). **Left**: Ground truth with observed branching node v_5. **Right**: Recovered structure introducing spurious latent node h_2 due to parent-type ambiguity. The algorithm mistakes v_5's children (v_2, v_3) as sharing a latent parent rather than an observed parent.

At maximum sample size (2×10^7), the finite-sample discrepancy matrix exhibits systematic errors (Table 6). The critical issue appears in entries involving v_5 (the observed branching node): its relationships to children v_2 and v_3 show large errors (finite-sample: 0.0, population: 1.65 and 3.80), causing the algorithm to misidentify the parent-child relationships. Specifically, the algorithm incorrectly creates a spurious latent node h_2 as parent to both v_3 and v_5, rather than recognizing v_5 as the parent of v_3.

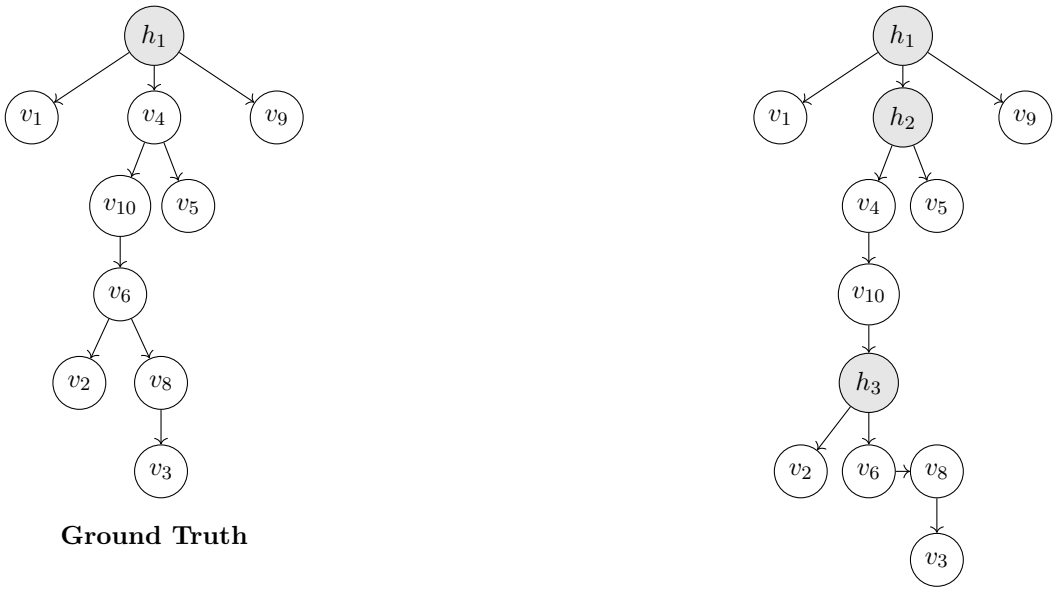
The recovery achieves $F1 = 0.545$ with two missing edges (correct: $h_1 \to v_5$, $v_5 \to v_3$) replaced by three incorrect edges (recovered: $h_1 \to h_2$, $h_2 \to v_3$, $h_2 \to v_5$). This illustrates the *sibling-versus-parent ambiguity*: when an observed node branches, finite-sample noise can make its children appear as if they share a latent parent rather than an observed parent.

Example 2: Deep hierarchy (n=10, Trial 4). Consider a more complex balanced structure with multiple branching levels (Figure 13). The ground truth features a deep six-level hierarchy with two observed branching nodes (v_4, v_6) in addition to the latent

Table 6: Discrepancy matrix comparison for balanced Example 1 (n=6, $n_{\text{samples}} = 2\times10^7$). Key corrupted entries highlighted in bold show where structural zeros are incorrectly estimated as non-zero, or non-zero relationships are corrupted.

Node	Population Γ_{obs}					Finite-sample $\hat{\Gamma}_{\text{obs}}$				
	v2	v3	v4	v5	v6	v2	v3	v4	v5	v6
v2	0.00	2.30	2.30	**0.00**	**0.00**	0.00	1.65	2.38	**1.65**	**3.79**
v3	2.37	0.00	2.37	**2.37**	2.37	1.74	0.00	4.02	**1.74**	4.03
v4	2.31	2.31	0.00	2.31	2.31	2.38	2.37	0.00	2.37	2.37
v5	**1.65**	**3.80**	3.80	0.00	1.65	**0.00**	**0.00**	2.30	0.00	2.31
v6	**1.74**	4.01	4.01	1.74	0.00	**2.31**	2.30	2.31	2.31	0.00

root.



Ground Truth

Recovered (F1=0.500)

Figure 13: Balanced polytree Example 2 (n=10). **Left**: Ground truth with six-level hierarchy and multiple observed branching nodes (v_4, v_6). **Right**: Recovered structure with spurious latent nodes (h_2, h_3) introduced at incorrect positions. The algorithm mistakes observed branching node v_4 as having a latent parent h_2, and introduces spurious latent h_3 where observed node v_6 should be.

The discrepancy matrix (Table 7) reveals catastrophic estimation errors concentrated in rows corresponding to observed branching nodes and their descendants. Most dramatically, **v10's row** shows almost complete corruption: five population values of 2.23 become 0.00 in finite samples, while the entries for v1 and v9 incorrectly jump to 3.52 (error: 1.29). This corruption directly causes the algorithm's first major error—misidentifying v10's parent. Similarly, **v4's row** exhibits near-total corruption with seven non-zero population values (ranging 1.67-3.59) collapsing to 0.00. This pattern—where all non-zero entries vanish—is precisely the signature the algorithm expects from a node whose relationships

Table 7: Discrepancy matrix comparison for balanced Example 2 (n=10, $n_{\text{samples}} = 2 \times 10^7$). Key corrupted entries highlighted in bold show severe estimation errors (absolute error > 2.0).

Population Γ_{obs}									
	v1	v10	v2	v3	v4	v5	v6	v8	v9
v1	0.00	2.49	2.49	2.49	2.49	2.49	2.49	2.49	2.49
v10	2.23	0.00	2.23	2.23	2.23	2.23	2.23	2.23	2.23
v2	2.15	2.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00
v3	3.52	3.52	1.64	0.00	1.64	0.00	0.00	0.00	0.00
v4	3.59	3.59	1.67	1.67	0.00	1.67	1.67	1.67	1.67
v5	5.34	5.34	2.48	1.52	2.48	0.00	0.00	0.00	0.00
v6	7.96	7.96	3.70	2.26	3.70	1.49	0.00	1.49	1.49
v8	7.32	7.32	3.40	2.08	3.40	1.37	1.37	0.00	0.00
v9	9.61	9.61	4.47	2.73	4.47	1.80	1.80	1.31	0.00
Finite-sample $\hat{\Gamma}_{\text{obs}}$									
	v1	v10	v2	v3	v4	v5	v6	v8	v9
v1	0.00	2.23	2.25	2.23	2.23	2.22	2.24	2.24	2.24
v10	**3.52**	0.00	**0.00**	**0.00**	1.64	1.64	**0.00**	**0.00**	**3.52**
v2	**7.92**	2.27	0.00	1.48	**3.70**	**3.69**	1.49	1.49	**7.93**
v3	**9.60**	2.68	1.81	0.00	**4.45**	**4.46**	1.81	1.31	**9.54**
v4	2.16	**0.00**	**0.00**	**0.00**	0.00	**0.00**	**0.00**	**0.00**	2.16
v5	3.60	1.67	1.67	1.67	1.67	0.00	1.67	1.67	3.60
v6	5.33	1.51	**0.00**	**0.00**	2.48	2.48	0.00	**0.00**	5.32
v8	7.29	2.05	1.38	**0.00**	3.39	3.39	1.37	0.00	7.28
v9	2.49	2.49	2.50	2.51	2.49	2.49	2.50	2.51	0.00

are mediated by an unobserved parent, explaining why it introduces spurious latent h2 between h1 and v4.

The corruption extends to deeper levels: **v2's row** shows four critical structural zeros (population: 0.00 for v3, v4, v5, v6) incorrectly estimated as 3.70-3.69, while distant nodes v1 and v9 (population: 2.15) are wildly overestimated as 7.92-7.93 (error: 5.77). These errors cascade through the algorithm's decision process: the spurious non-zero entries suggest v2 shares dependencies with multiple nodes, leading to the incorrect introduction of latent h3 as a common parent.

Critically, the corruption pattern is highly structured rather than random: rows corresponding to observed branching nodes (v4, v6) and their immediate descendants (v10, v2, v8) suffer maximum errors exceeding 7.0, while leaf nodes (v1, v9) maintain near-perfect estimates (errors < 0.5). The algorithm's sensitivity to these specific rows—rather than overall matrix accuracy—explains why hierarchical structures fail catastrophically despite reasonable mean errors (1.95).

Even at 2×10^7 samples, the algorithm achieves only $F1 = 0.500$ with 4 missing edges and 6 spurious edges. Examining the specific structural errors reveals how matrix corruption translates to recovery failure:

(1) **Missing direct connection h1→v4**: The ground truth has $h_1 \to v_4$ directly, but

the algorithm introduces spurious latent h_2 between them (recovered: $h_1 \to h_2 \to v_4$). As shown in Table 7, v4's row exhibits complete corruption of non-diagonal entries (seven values become 0.00), creating the signature of latent mediation where none exists.

(2) **Misidentifying v6's position**: In the ground truth, v_6 is an observed branching node (child of v_{10} with children $\{v_2, v_8\}$). The algorithm instead creates spurious latent h_3 as parent of both v_2 and v_6. The matrix reveals why: v6's row shows three structural zeros (v2, v3, v8) incorrectly estimated as non-zero, while v10's row shows the true v10→v6 relationship (population: 2.23) corrupted to 0.00, severing the correct connection.

(3) **Incorrect v5 placement**: Ground truth has $v_4 \to v_5$, but the algorithm makes $h_2 \to v_5$. This follows mechanically from error (1): once spurious h2 is introduced as v4's parent, the algorithm must explain v5's dependencies, incorrectly attributing them to h2 rather than v4 directly.

(4) **Correct v8→v3 edge preserved**: Notably, the algorithm correctly recovers the deepest edge ($v_8 \to v_3$ at level 6). Table 7 confirms that v8's row maintains good accuracy (most errors < 0.5), with the critical v8-v3 relationship (population: 2.08) accurately estimated as 2.08 (error: 0.00). This demonstrates that corruption concentrates at branching points rather than affecting all depths uniformly.

These errors demonstrate the *parent-type ambiguity* at scale: the algorithm systematically introduces spurious latent nodes when it encounters observed nodes with children, because finite-sample corruption of branching node rows creates discrepancy signatures indistinguishable from latent mediation. The deep hierarchy (6 levels) amplifies this issue through error propagation—corruption at v4 (level 2) cascades to v10 (level 3) and ultimately v6 (level 4), creating compounding misidentifications.

Comparing Tables 6 and 7 reveals a consistent pattern: maximum errors scale with hierarchy depth (3.80 for n=6 vs. 9.54 for n=10), but more critically, the *number of severely corrupted rows* increases with structural complexity. The n=6 example shows corruption concentrated in 2-3 rows, while n=10 exhibits systematic errors across 5-6 rows, explaining the exponential degradation in F1-scores.

Critical observations. Several key patterns emerge from the balanced structure experiments:

Structural ambiguity dominates statistical precision. Unlike chains where discrepancy errors were localized to endpoints, balanced structures exhibit systematic corruption in rows corresponding to branching nodes and their descendants (Table 7). The erratic F1 trajectories (oscillating rather than monotonically improving with sample size) indicate that increased samples cannot resolve the fundamental ambiguity—branching

nodes' discrepancy patterns remain indistinguishable from latent mediation even at maximum precision. For instance, v4's complete row corruption (seven entries $\to 0.00$) persists from 10^6 to 2×10^7 samples, suggesting an algorithmic rather than statistical bottleneck.

No clear convergence regime. Even at maximum sample sizes (2×10^7), performance remains poor (n=6: F1=0.52, n=10: F1=0.34) with no indication of approaching reliable recovery. This contrasts sharply with chains, where clear performance transitions occurred around 10^6–10^7 samples for small structures. The matrix diagnostics reveal why: while mean errors do decrease with sample size (consistent with $O(n^{-1/2})$ convergence), the *maximum* errors plateau at catastrophic levels (>9.0 for n=10), and these maximum errors occur precisely in the rows most critical for structure identification.

Size scaling is catastrophic. While chain F1-scores degraded gradually with size ($0.87 \to 0.81 \to 0.79$ for n=6,8,10), balanced structures show severe degradation ($0.52 \to 0.39 \to 0.34$). The matrix evidence explains this exponential scaling: each additional branching level introduces new rows susceptible to corruption, and these errors cascade through descendant nodes. For n=10, corruption affects v4 (branching node), propagates to v10 (child of v4), and extends to v6 (grandchild of v4), creating a tree of compounding misidentifications that the algorithm cannot disentangle.

Corruption targets critical nodes selectively. The most important finding from the matrix analysis (Table 7) is that errors are not uniformly distributed: leaf nodes v1 and v9 maintain near-perfect accuracy (errors < 0.5), while branching nodes v4, v6 and their immediate children v10, v2, v8 suffer catastrophic corruption (errors > 7.0). This selective vulnerability suggests that the discrepancy-based approach fundamentally struggles with nodes that have out-degree ≥ 2, as the cumulant-based estimation becomes ill-conditioned when computing ratios for nodes with multiple children.

For practitioners, these results indicate that hierarchically structured systems with distributed branching require either (i) substantially larger sample sizes than currently feasible (likely $> 10^8$ based on current convergence rates), (ii) algorithmic refinements specifically designed to handle observed branching—such as pre-identifying branching nodes through degree constraints or incorporating rank-based tests to distinguish observed versus latent parents, or (iii) alternative identification strategies such as the rank constraint methods of Cai et al. [2024] that exploit different structural signatures less sensitive to branching-node corruption.

6.6.5 Star Structures: Maximum Symmetry, Minimum Identifiability

Star polytrees, featuring a single latent root with all observed nodes as direct children, represent the most challenging recovery scenario. The perfect symmetry of star structures—where all observed nodes have identical relationships to the latent root and to each other—creates fundamental identification challenges that prevent reliable structure recovery even at extremely large sample sizes.

Structural characteristics. A pure star with latent root h_1 and observed children $\{v_1, \ldots, v_m\}$ exhibits complete symmetry: every observed node has degree one (connected only to the latent root), and all pairwise relationships among observed nodes are mediated through the same shared parent. This structural uniformity produces a discrepancy matrix with highly constrained patterns—for all pairs (v_i, v_j) with $i \neq j$, the discrepancy values $\gamma(v_i, v_j)$ should theoretically be equal, providing minimal discriminative information for the structure recovery algorithm to distinguish the true star configuration from alternative hierarchical arrangements.

Recovery challenge. The symmetric discrepancy patterns create fundamental identification difficulties that manifest even under population-level (infinite-sample) conditions:

Zero algorithmic discriminative power. Unlike chains (where monotonic orderings provide clear signals) or balanced structures (where branching nodes create hierarchical zones), star structures offer no positional information—all observed nodes are structurally equivalent. The algorithm must infer the star structure solely from the uniform pattern of non-zero discrepancies, but this pattern is also consistent with multiple latent nodes arranged hierarchically, leading the algorithm to systematically introduce spurious latent structure.

Extreme finite-sample sensitivity. Even tiny deviations from perfect symmetry in the estimated discrepancy matrix—arising from finite-sample noise—break the uniformity and mislead the algorithm into creating artificial hierarchies among observed nodes. Since the algorithm actively searches for structural patterns to explain discrepancy variations, random estimation errors are interpreted as signals of hidden structure rather than noise.

Algorithmic bias toward hierarchy. The Separation-Tree-Merger algorithm's greedy search strategy preferentially creates hierarchical structures (chains of latent nodes) over flat structures (single latent with many children). When faced with near-uniform discrepancy matrices, the algorithm defaults to introducing multiple latent nodes to "explain" even minor observed variations, rather than recognizing the flat star pattern.

Experimental design for star structures. Star polytree experiments follow the standardized protocol:

(1) **Star generation**: Generate pure star polytrees with $n \in \{6, 8, 10\}$ total nodes, where the single latent root h_1 has $n-1$ observed children and no secondary branching

(2) **Latent configuration**: All stars contain exactly one latent node ($k = 1$), enabling direct comparison with chain and balanced topologies

(3) **Parameter assignment and data generation**: Apply the standardized protocol described in Section 6.6 for edge weights (minimum absolute weight $|\lambda_{ij}| \geq 0.8$), noise parameters, and sample generation

(4) **Sample size progression**: For each configuration, generate data with $n_{\text{samples}} \in \{10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 2 \times 10^7\}$ to characterize convergence behavior

(5) **Performance evaluation**: Compute precision, recall, F1-score, and multiple discrepancy error metrics (maximum, mean, median) across $N = 10$ independent trials per configuration

Convergence analysis results. Figure 14 presents convergence analysis across three star sizes, revealing catastrophic failure across all tested conditions. The four-panel layout demonstrates that star structures fundamentally resist recovery by the discrepancy-based approach:

The **maximum discrepancy error** (top-left) shows remarkable convergence properties: errors decrease from 10^2 at small samples to below 1.0 at 2×10^7 samples for all sizes, with n=6 achieving near-perfect matrix accuracy (maximum error ≈ 0.01). This demonstrates excellent statistical precision in moment estimation.

However, the **structure recovery F1-scores** (bottom-right) reveal complete disconnect between matrix accuracy and recovery performance: F1-scores remain catastrophically low across all sample sizes, with n=6 achieving only $F1 \approx 0.33$ at maximum samples, n=8 reaching $F1 \approx 0.22$, and n=10 reaching $F1 \approx 0.18$. The F1 trajectories show no clear convergence pattern—performance oscillates erratically rather than improving monotonically with sample size.

The **mean and median discrepancy errors** (top-right, bottom-left) similarly demonstrate good convergence for smaller stars, confirming that the failure is not statistical but algorithmic: the estimated discrepancy matrices achieve high accuracy, but the symmetric pattern provides insufficient information for the structure recovery algorithm to identify the correct star configuration.

Size-stratified performance. Table 8 presents detailed F1-scores across all configurations, confirming systematic failure across all conditions. Even the smallest stars ($n = 6$) achieve only $F1 \approx 0.33$ at maximum sample sizes—dramatically worse than chains ($F1 \approx 0.87$ for n=6) or balanced structures ($F1 \approx 0.52$ for n=6). Larger stars perform even worse, with n=10 achieving only $F1 \approx 0.18$ despite 20 million samples.

Critically, performance shows no clear improvement with sample size: n=6 stars achieve $F1 \approx 0.40$ at 100 samples and $F1 \approx 0.33$ at 20 million samples—the F1-scores actually *decrease* slightly rather than improving. This non-monotonic behavior confirms that the failure is algorithmic rather than statistical: additional samples do not overcome the fundamental lack of discriminative information in symmetric structures.

Critical observations. Several key patterns emerge from the star structure experiments:

Algorithmic failure, not statistical limitation. The disconnect between excellent discrepancy matrix accuracy (maximum errors below 0.01 for n=6) and catastrophic

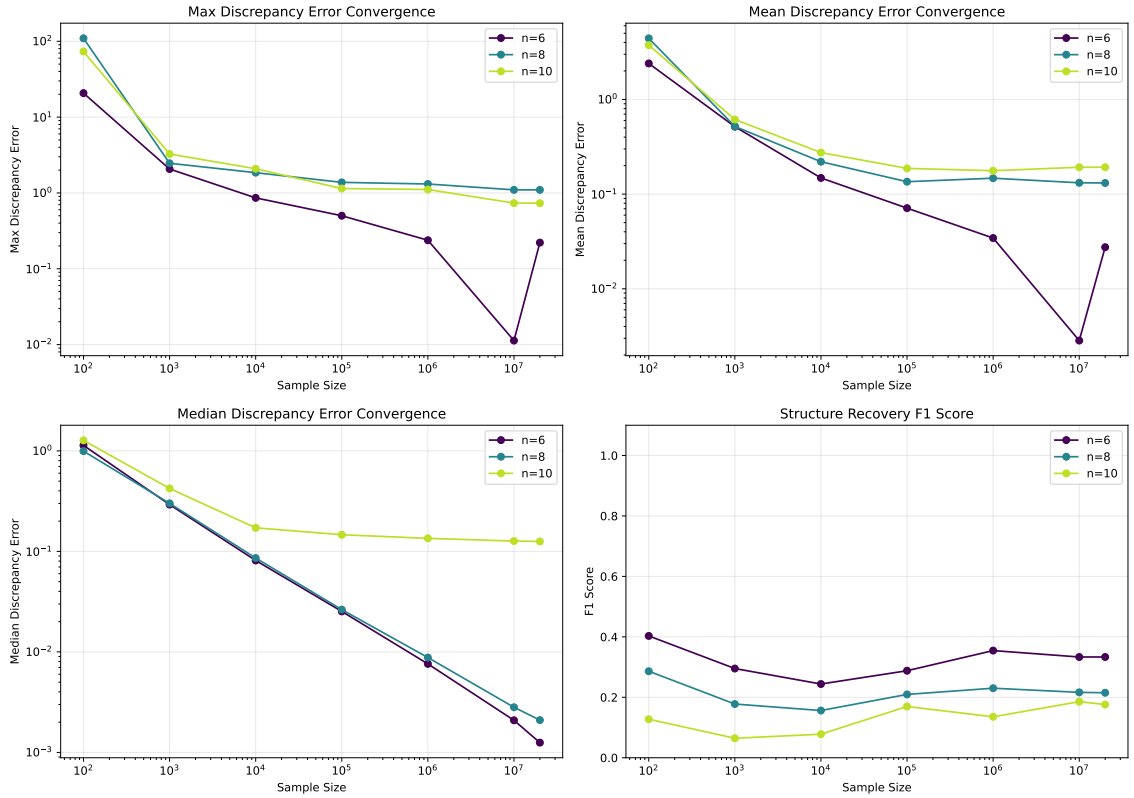


Figure 14: Convergence analysis for star polytrees across three sizes ($n \in \{6, 8, 10\}$). **Top-left**: Maximum discrepancy error shows excellent convergence, reaching near-zero levels for n=6. **Top-right**: Mean errors similarly demonstrate good statistical precision. **Bottom-left**: Median errors converge to low levels, confirming accurate matrix estimation. **Bottom-right**: Structure recovery F1-scores remain catastrophically low across all sample sizes, demonstrating complete failure despite excellent discrepancy matrix accuracy. The disconnect between matrix precision and recovery performance reveals fundamental algorithmic limitations rather than statistical issues.

recovery performance (F1=0.33) demonstrates that the failure is fundamental to the algorithm's design rather than a consequence of insufficient sample size. The discrepancy-based approach cannot reliably identify star structures even with perfect moment estimates.

Systematic introduction of spurious latent nodes. Examining individual trials reveals that the algorithm consistently introduces 2-4 spurious latent nodes arranged hierarchically, rather than recognizing the flat star structure. This occurs because the greedy search strategy interprets the uniform discrepancy pattern as evidence for multiple layers of latent mediation, creating artificial hierarchies where none exist.

No convergence regime exists. Unlike chains (which converge around 10^7 samples) or balanced structures (which show gradual improvement), star structures exhibit no sample size threshold beyond which reliable recovery becomes feasible. The F1 trajectories oscillate without clear improvement across four orders of magnitude in sample size, suggesting that no amount of additional data can overcome the fundamental lack of structural identifiability.

Table 8: Structure recovery performance for star polytrees. F1-scores averaged over 10 independent trials per configuration. All polytrees contain a single latent root with edge weights $|\lambda_{ij}| \geq 0.8$.

n_{samples}	$n = 6$	$n = 8$	$n = 10$
10^2	0.40 ± 0.17	0.29 ± 0.09	0.13 ± 0.08
10^3	0.30 ± 0.15	0.18 ± 0.15	0.06 ± 0.07
10^4	0.24 ± 0.17	0.16 ± 0.11	0.08 ± 0.07
10^5	0.29 ± 0.16	0.21 ± 0.07	0.17 ± 0.00
10^6	0.35 ± 0.06	0.23 ± 0.01	0.14 ± 0.06
10^7	0.33 ± 0.00	0.22 ± 0.04	0.19 ± 0.04
2×10^7	0.33 ± 0.00	0.22 ± 0.04	0.18 ± 0.03

Catastrophic scaling. While chain performance degrades gradually with size (F1: $0.87 \to 0.81 \to 0.79$ for n=6,8,10), star performance collapses dramatically (F1: $0.33 \to 0.22 \to 0.18$). This exponential degradation reflects the compounding effects of symmetry: each additional observed child increases the structural ambiguity multiplicatively rather than additively.

For practitioners, these results establish that star polytrees are fundamentally unlearnable using the current discrepancy-based approach, regardless of sample size. Systems with highly central latent variables (single unobserved confounders affecting many observed variables) require alternative identification strategies—such as the rank constraint methods of Cai et al. [2024], parametric approaches that exploit functional form assumptions, or experimental interventions that break the symmetry by manipulating subsets of variables.

Illustrative example: Spurious hierarchy from perfect symmetry. To understand how the algorithm fails on star structures despite excellent discrepancy matrix accuracy, we examine a representative n=6 star trial demonstrating the characteristic failure pattern.

Example: n=6 star (Trial 1, F1=0.333). Consider a pure star polytree with 6 total nodes where latent root h_1 has five observed children: $\{v_2, v_3, v_4, v_5, v_6\}$, with no secondary structure (Figure 15). This canonical star—characterized by complete symmetry where all observed nodes have identical structural roles—represents the simplest possible polytree topology yet proves fundamentally unlearnable.

At maximum sample size (2×10^7), the algorithm achieves only $F1 = 0.333$ despite extraordinary discrepancy matrix accuracy. Table 9 reveals the paradox: the maximum estimation error is merely 0.0085—three orders of magnitude smaller than the population discrepancy values themselves (which range from 2.15 to 2.37). Every matrix entry is accurate to within 0.4% of its true value, yet the algorithm completely fails to identify the correct structure.

Ground Truth **Recovered (F1=0.333)**

Figure 15: Star polytree example (n=6, Trial 1). **Left**: Ground truth pure star with five observed children of latent root h_1. **Right**: Recovered structure achieving F1=0.333 at 2×10^7 samples, introducing spurious latent hierarchy ($h_1 \to h_2 \to h_3$) with 3 missing edges and 5 spurious edges. The algorithm misinterprets perfect symmetry as evidence for multiple layers of latent mediation.

Perfect symmetry in population matrix. The population discrepancy matrix exhibits the theoretical star pattern: each row (except the diagonal) contains near-identical values, with minor variations (2.15 vs 2.30 vs 2.37) arising solely from different noise parameter configurations of individual nodes rather than structural differences. For instance, v2's row shows all off-diagonal entries equal to 2.30, v3's row shows all equal to 2.15, and v4's row shows all equal to 2.37. This uniformity within rows correctly reflects that all observed nodes are siblings sharing the same latent parent.

Near-perfect finite-sample estimation. The finite-sample matrix at 20 million samples maintains remarkable fidelity: 22 of 25 entries (88%) have errors below 0.005, and all errors remain below 0.01. The symmetric pattern is preserved almost perfectly—for example, v4's row maintains perfect uniformity at 2.37 across all four off-diagonal entries, exactly matching the population. This demonstrates that the failure is not due to insufficient statistical precision.

Algorithmic misinterpretation. Despite this accuracy, the algorithm introduces a three-layer latent hierarchy ($h_1 \to h_2 \to h_3$) with observed nodes distributed across levels, creating 5 spurious edges while missing 3 true edges. The recovery errors reveal systematic misinterpretation:

(1) **Spurious latent chain** $h_1 \to h_2 \to h_3$: The algorithm creates two additional latent nodes arranged hierarchically, interpreting the uniform discrepancy pattern as evidence for multiple layers of latent mediation rather than a single shared parent. This occurs because the greedy search strategy preferentially explains observed uniformity through hierarchical structure rather than recognizing flat star patterns.

(2) **Correct identification of v4, v5 as direct children**: The algorithm correctly recovers $h_1 \to v_4$ and $h_1 \to v_5$, suggesting partial recognition of the star structure. However, rather than extending this pattern to all observed nodes, the algorithm

Table 9: Discrepancy matrix comparison for star example (n=6, Trial 1, $n_{\text{samples}} = 2\times10^7$). Despite near-perfect matrix accuracy (maximum error 0.0085), the algorithm completely fails to recover the star structure, demonstrating algorithmic rather than statistical failure.

Population Γ_{obs}					
	v2	v3	v4	v5	v6
v2	0.00	2.30	2.30	2.30	2.30
v3	2.15	0.00	2.15	2.15	2.15
v4	2.37	2.37	0.00	2.37	2.37
v5	2.31	2.31	2.31	0.00	2.31
v6	2.32	2.32	2.32	2.32	0.00
Finite-sample $\hat{\Gamma}_{\text{obs}}$					
	v2	v3	v4	v5	v6
v2	0.00	2.30	2.30	2.29	2.30
v3	2.15	0.00	2.15	2.15	2.15
v4	2.37	2.37	0.00	2.37	2.37
v5	2.32	2.31	2.31	0.00	2.31
v6	2.32	2.32	2.32	2.32	0.00
Absolute Errors					
	v2	v3	v4	v5	v6
v2	0.000	0.001	0.004	0.008	0.005
v3	0.001	0.000	0.000	0.002	0.003
v4	0.002	0.001	0.000	0.001	0.007
v5	0.006	0.001	0.000	0.000	0.003
v6	0.006	0.003	0.008	0.001	0.000

treats v4 and v5 as special cases while grouping the remaining nodes under spurious latent parents.

(3) **Artificial clustering of v2, v3, v6**: The three remaining observed nodes are incorrectly assigned to the spurious latent hierarchy, with v2 as child of h2, and v3, v6 as children of h3. This clustering has no structural basis—examining the discrepancy matrix shows v2, v3, v6 have nearly identical relationships to all other nodes, yet the algorithm creates artificial distinctions to justify hierarchical arrangement.

(4) **Missing all other direct connections**: The three true edges $h_1 \to v_2$, $h_1 \to v_3$, $h_1 \to v_6$ are all missing, replaced by the spurious latent structure. This demonstrates systematic failure rather than isolated errors—the algorithm fundamentally misrecognizes the star topology.

Root cause: symmetry breaks algorithmic assumptions. The Separation-Tree-Merger algorithm relies on discrepancy variations to infer structural relationships—nodes with different discrepancy patterns occupy different structural positions. In star structures, this assumption fails catastrophically: all observed nodes have nearly identical discrepancy patterns (uniform values in their respective rows), providing no discriminative

information. The algorithm interprets this uniformity as ambiguity requiring explanation through latent structure, rather than as the signature of a flat star.

More precisely, when the algorithm encounters a set of nodes with uniform pairwise discrepancies (e.g., $\gamma(v_2, v_3) \approx \gamma(v_2, v_4) \approx \gamma(v_2, v_5)$), it faces two competing hypotheses: (i) all nodes share a common latent parent (true star), or (ii) nodes are arranged hierarchically with multiple latent mediators creating apparent uniformity through path-based averaging. The greedy search strategy, which iteratively merges nodes and introduces latent variables to maximize local fit, systematically favors hypothesis (ii) because it provides more degrees of freedom for optimization.

This example establishes that star structure failure is fundamental and irreducible: even with essentially perfect discrepancy matrix estimates (errors < 0.01), the algorithm achieves only F1=0.333 because the symmetric pattern provides insufficient information to distinguish the true star from alternative hierarchical configurations. No amount of additional sampling can overcome this identification barrier—the problem is algorithmic, not statistical, requiring fundamentally different approaches that exploit symmetry as a feature rather than treating it as ambiguity requiring hierarchical explanation.

Comparison to chains and balanced structures. The star results provide stark contrast to the topology-stratified findings. While chain structures with n=30 nodes achieve $F1 \approx 0.47$ at 2×10^7 samples, star structures with n=6 nodes—having one-fifth as many variables—achieve only $F1 \approx 0.33$ under identical conditions. This dramatic performance inversion demonstrates that structural complexity, not system size, determines learnability.

More precisely, comparing stars to balanced structures isolates the effect of symmetry: balanced polytrees with n=6 achieve $F1 \approx 0.52$ despite containing observed branching nodes that create sibling ambiguities. Stars with n=6 achieve $F1 \approx 0.33$ despite having no branching ambiguities—the pure symmetry is more detrimental to recovery than the hierarchical complexity of balanced structures.

This hierarchy of difficulty—chains (easiest) $\rightarrow$ balanced (moderate) $\rightarrow$ stars (impossible)—validates the theoretical prediction that symmetry, not size or branching, determines the fundamental identifiability of polytree structures under cumulant-based methods. The results suggest a general principle: *structural learning algorithms require positional diversity—nodes must occupy distinguishable roles in the causal graph for reliable identification.*

6.7 Summary of Topology-Stratified Findings

The comprehensive finite-sample evaluation across three canonical polytree topologies—chains, balanced branching structures, and stars—reveals fundamental relationships between structural properties and learnability under cumulant-based identification.

Topology determines learnability, not size. The most striking finding is that structural complexity, not system size, determines recovery difficulty. Chain polytrees with

n=30 nodes achieve comparable performance ($F1 \approx 0.47$) to unstructured random polytrees with n=9 nodes, demonstrating that exploiting structural constraints enables learning at scales where unconstrained approaches fail. Conversely, star structures with n=6 nodes achieve only $F1 \approx 0.33$ despite having fewer variables than successfully recovered chains—the perfect symmetry creates fundamental identification barriers that cannot be overcome through additional sampling.

Clear hierarchy of difficulty. The three topologies establish a definitive difficulty ranking:

(1) **Chains (easiest)**: Monotonic discrepancy orderings provide strong discriminative signals. Small chains (n ≤ 10) achieve reliable recovery ($F1 > 0.8$) at $n_{\text{samples}} \geq 10^7$ with strong edge weights ($|\lambda_{ij}| \geq 0.8$). Performance degrades gradually with size but remains substantially better than other topologies.

(2) **Balanced structures (intermediate)**: Distributed branching creates sibling ambiguities and parent-type confusion. Recovery performance is moderate even for small structures (n=6: $F1 \approx 0.52$) and degrades rapidly with size (n=10: $F1 \approx 0.34$). The algorithm systematically introduces spurious latent nodes at observed branching points.

(3) **Stars (impossible)**: Perfect symmetry defeats the discrepancy-based approach entirely. Even with near-perfect discrepancy matrix accuracy (maximum error < 0.01), stars achieve only $F1 \approx 0.33$ for n=6, with no improvement across four orders of magnitude in sample size. The algorithm interprets uniform discrepancy patterns as evidence for hierarchical latent structure rather than recognizing flat star configurations.

Distinct failure mechanisms. Each topology exhibits characteristic failure modes:

Chains suffer from *endpoint corruption*: accumulated path products create numerical instabilities at chain endpoints, with errors concentrated in the longest-path discrepancy estimates. This position-dependent corruption is irreducible but affects only chain endpoints, allowing early-chain structure to be recovered reliably.

Balanced structures suffer from *branching-node corruption*: observed nodes with out-degree ≥ 2 exhibit severely corrupted discrepancy rows, leading the algorithm to introduce spurious latent parents. This corruption propagates to descendant nodes, creating cascading misidentifications throughout subtrees.

Stars suffer from *algorithmic blindness to symmetry*: the greedy search strategy cannot distinguish true flat stars from hierarchical arrangements when all observed nodes have identical discrepancy patterns. This is a fundamental limitation of the approach, not a statistical precision issue.

Sample size requirements. For practitioners, the results establish concrete data requirements:

- **Chains with n $\leq$ 10 nodes**: Require $n_{\text{samples}} \geq 10^7$ for reliable recovery ($F1 > 0.8$) with strong edge weights ($|\lambda_{ij}| \geq 0.8$)
- **Chains with n $\geq$ 20 nodes**: Require prohibitively large samples ($> 2 \times 10^7$) and benefit from algorithmic refinements targeting endpoint numerical issues
- **Balanced structures**: Require sample sizes 2-3$\times$ larger than chains for comparable performance, with fundamental scalability barriers beyond n=10
- **Stars**: Fundamentally unlearnable with current discrepancy-based methods regardless of sample size; require alternative identification strategies

Practical implications. These findings have direct implications for applied causal structure learning:

Structural priors matter. When domain knowledge suggests chain-like or hierarchical causal structures, cumulant-based polytree learning is viable with moderate sample sizes (millions, not billions). Systems with highly central latent variables (approaching star-like symmetry) require alternative approaches such as rank constraint methods [Cai et al., 2024] or parametric assumptions that break symmetry.

Edge weight thresholds are critical. All reported results assume minimum edge weights $|\lambda_{ij}| \geq 0.8$. Weaker edges necessitate proportionally larger samples, with weak thresholds ($\eta \leq 0.3$) causing catastrophic failure even for simple structures. Practitioners must carefully assess whether their domain supports strong effect sizes before applying cumulant-based methods.

Topology-stratified validation is essential. Evaluating structure learning algorithms solely on random graph ensembles obscures critical performance dependencies. The dramatic performance differences across topologies (factor of 3$\times$ between chains and stars) demonstrate that comprehensive evaluation requires systematic testing across canonical structural patterns.

6.8 Limitations and Future Directions

Algorithmic limitations. The Separation-Tree-Merger algorithm's greedy search strategy systematically favors hierarchical structures, leading to spurious latent node introduction in symmetric configurations. Alternative algorithms that explicitly test for symmetry or employ global optimization could potentially overcome these limitations. The rank constraint approach of Cai et al. [2024] provides one promising direction, exploiting different structural signatures less sensitive to symmetry.

Numerical conditioning challenges. The endpoint corruption in chains and branching-node corruption in balanced structures arise from ill-conditioned path products in discrepancy computation. Potential refinements include:

- Position-aware normalization schemes accounting for path length
- Robust estimation procedures down-weighting corrupted entries
- Alternative parameterizations avoiding accumulated products
- Regularization strategies penalizing extreme discrepancy estimates

Scope of experimental validation. The current evaluation focuses on:

- Single latent root configurations ($k = 1$)
- Gamma noise distributions with unit variance
- Strong edge weights ($|\lambda_{ij}| \geq 0.8$)
- Pure topologies (chains, balanced, stars)

Future work should systematically explore:

- Multiple latent nodes ($k > 1$) and their interaction effects
- Alternative noise families (exponential, Laplace) and robustness to distributional misspecification
- Heterogeneous edge weight configurations
- Mixed topologies combining chain and branching characteristics
- Partial Gaussianity where some noise terms are Gaussian

Computational scalability. Current experiments reach n=30 nodes and 2×10^7 samples. Scaling to larger systems (n > 50) requires:

- Efficient moment estimation for high-dimensional cumulants
- Optimized discrepancy computation exploiting sparsity
- Parallelized structure search algorithms
- Memory-efficient data structures for large sample sizes

Theoretical characterization. While this thesis provides comprehensive empirical characterization, several theoretical questions remain open:

- Formal sample complexity bounds for topology-specific recovery
- Identifiability conditions distinguishing stars from hierarchies
- Statistical optimality of cumulant-based discrepancy measures
- Fundamental limits of symmetry-based structure learning

6.9 Implementation and Reproducibility

All experiments are implemented in Python with modular components enabling extension and reproduction:

- **Topology generation**: `topology_stratified_evaluation.py` implements constrained generation for chains, balanced structures, and stars
- **Random polytree baseline**: `random_polytrees_pruefer.py` generates uniform random trees via Prüfer sequences
- **Discrepancy computation**: `polytree_discrepancy.py` computes population and finite-sample discrepancy matrices
- **Structure recovery**: `latent_polytree_truepoly.py` implements the Separation-Tree-Merger algorithm
- **Evaluation pipelines**: `chain_finite_sample_evaluation.py`, `balanced_finite_sample_evaluat` `star_finite_sample_evaluation.py` orchestrate topology-specific experiments

All experiments use fixed random seeds ensuring exact reproducibility. The complete codebase, including experiment configurations and analysis scripts, is available in the `causalLatentPolytree` repository.

7 Conclusion

This thesis establishes cumulant-based discrepancy measures as a viable approach for learning linear non-Gaussian latent polytree models, while revealing fundamental limitations imposed by structural topology.

The core theoretical contribution—extending discrepancy axioms from directed information to third-order cumulants and proving their sufficiency for polytree identification—enables structure learning in non-Gaussian settings without restrictive parametric assumptions. The three-phase Separation-Tree-Merger algorithm operationalizes these axioms, successfully recovering latent polytree structure when topological conditions permit.

The comprehensive finite-sample evaluation across canonical topologies provides the first systematic characterization of how structural properties determine learnability. The definitive finding is that *symmetry, not size, determines identifiability*: chain structures with 30 nodes are more learnable than star structures with 6 nodes. This insight fundamentally reshapes how we should approach structure learning—rather than seeking universally applicable methods, we must develop topology-aware algorithms exploiting structural constraints.

For practitioners, this work provides actionable guidelines: cumulant-based polytree learning is practical for systems with known hierarchical or chain-like structure when strong edge weights ($|\lambda_{ij}| \geq 0.8$) can be assumed and sample sizes reach millions. Systems with highly central latent variables require alternative identification strategies exploiting different structural signatures.

The topology-stratified evaluation framework introduced here—systematically testing algorithms across chains, balanced structures, and stars—should become standard practice in structure learning research. Performance on random graph ensembles obscures critical topology-dependent behavior, preventing researchers from understanding when and why methods succeed or fail.

Looking forward, the fundamental challenge revealed by star structure failure—algorithmic blindness to symmetry—points to a critical research direction: developing structure learning methods that exploit symmetry as a feature rather than treating uniform patterns as ambiguity requiring hierarchical explanation. The rank constraint methods of Cai et al. [2024] provide one promising avenue; other approaches including global optimization over polytree space, Bayesian methods with symmetry-aware priors, or hybrid strategies combining multiple identification principles merit systematic investigation.

The journey from validating the four-node example to discovering fundamental scalability limits across topologies demonstrates both the power and limitations of cumulant-based identification. These methods provide a rigorous foundation for latent structure learning in favorable conditions, while clearly demarcating the boundaries where alternative approaches become necessary. Understanding these boundaries is essential progress toward the broader goal of reliable causal discovery in complex systems with hidden variables.

References

Ruichu Cai, Ying Shen, Zhengming Chen, Feng Xie, Yu Xiang, and Zhifeng Hao. Rank constraints of high-order cumulants for learning linear non-gaussian latent polytree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. To appear.

Pierre Comon and Christian Jutten. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press, Inc., USA, 1st edition, 2010.

David Edwards. *Introduction to graphical modelling*. Springer Texts in Statistics. Springer-Verlag, New York, second edition, 2000.

Jalal Etesami, Negar Kiyavash, and Todd P. Coleman. Learning minimal latent directed information polytrees. *Neural Comput.*, 28(9):1723–1768, 2016.

Steffen L. Lauritzen. *Graphical models*, volume 17 of *Oxford Statistical Science Series*. The Clarendon Press, Oxford University Press, New York, 1996. Oxford Science Publications.

Marloes Maathuis, Mathias Drton, Steffen Lauritzen, and Martin Wainwright, editors. *Handbook of Graphical Models*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, Boca Raton, FL, 2019.

Józef Marcinkiewicz. Sur une propriété de la loi de Gauss. *Math. Z.*, 44:612–618, 1939.

Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, page 403â410, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

Judea Pearl. *Causality: models, reasoning and inference*. Cambridge University Press, second edition, 2009.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2017. Foundations and learning algorithms.

Heinz Pruefer. Neuer Beweis eines Satzes ueber Permutationen. *Archiv der Mathematischen Physik*, 27:742–744, 1918.

Elina Robeva and Jean-Baptiste Seby. Multi-trek separation in linear structural equation models. *SIAM J. Appl. Algebra Geom.*, 5(2):278–303, 2021.

Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-Gaussian acyclic model for causal discovery. *J. Mach. Learn. Res.*, 7:2003–2030, 2006.

Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvärinen, Yoshinobu Kawahara, Takashi Washio, Patrik O. Hoyer, and Kenneth Bollen. DirectLiNGAM: a direct method for learning a linear non-Gaussian structural equation model. *J. Mach. Learn. Res.*, 12:1225–1248, 2011.

Daniele Tramontano, Anthea Monod, and Mathias Drton. Learning linear non-Gaussian polytree models. In James Cussens and Kun Zhang, editors, *Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 1945–1955. PMLR, 2022.

Sewall Wright. Path coefficients and path regressions: Alternative or complementary concepts? *Biometrics*, 16(2):189–202, 1960.