# Robust Model Selection of Gaussian Graphical Models

**Anonymous authors**
**Paper under double-blind review**

## Abstract

In Gaussian graphical model selection, noise-corrupted samples present significant challenges. It is known that even minimal amounts of noise can obscure the underlying structure, leading to fundamental identifiability issues. A recent line of work addressing this "robust model selection" problem narrows its focus to tree-structured graphical models. Even within this specific class of models, exact structure recovery is shown to be impossible. However, several algorithms have been developed that are known to provably recover the underlying tree-structure up to an (unavoidable) equivalence class.

In this paper, we extend these results beyond tree-structured graphs. We first characterize the equivalence class up to which general graphs can be recovered in the presence of noise. Despite the inherent ambiguity (which we prove is unavoidable), the structure that can be recovered reveals local clustering information and global connectivity patterns in the underlying model. Such information is useful in a range of real-world problems, including power grids, social networks, protein-protein interactions, and neural structures. We then propose an algorithm which provably recovers the underlying graph up to the identified ambiguity. We further provide finite sample guarantees in the high-dimensional regime for our algorithm and validate our results through numerical simulations.

## 1 Introduction

Probabilistic graphical models have emerged as a powerful and flexible formalism for expressing and leveraging relationships among entities in large interacting systems (Lauritzen, 1996). They have found application in a range of areas including signal processing (Kim & Smaragdis, 2013; Ott & Stoop, 2006; Murphy et al., 2013), power systems (Anguluri et al., 2022; Deka et al., 2020; 2015), (phylo)genomics (Zuo et al., 2017; Dasarathy et al., 2014; 2022), and neuroscience (Bullmore & Bassett, 2011; Vinci et al., 2019). Gaussian graphical models are an important subclass of graphical models and are the main focus of this paper; our techniques do apply more broadly, as discussed in Section 7.

In several applications, we do not know the underlying graph structure, and the goal is to learn this from data — a problem dubbed graphical model selection. This is important because the graph structure provides a succinct representation of the complex multivariate distribution and can reveal important relationships among the underlying variables. See, e.g., Drton & Maathuis (2017); Maathuis et al. (2018) and references therein for more on this problem. Here, we focus on a relatively new but important task where samples from the underlying distribution are corrupted by independent noise with unknown variances. This occurs in a wide variety of applications where sensor data or experimental measurements suffer from statistical uncertainty or measurement noise. In these situations, we refer to the task of graph structure learning as *robust model selection*.

This problem was recently considered by Katiyar et al. (2019) and a line of follow-up work (Katiyar et al., 2020; Casanellas et al., 2021; Tandon et al., 2021) who show that unfortunately the conditional independence structure of the underlying distribution can be completely lost in general under such corruption; see Section 3.1 for more on this. The authors show that even in the often tractable case of tree-structured graphical models, one can only identify the structure up to an equivalent class. In fact, the assumption that the underlying uncorrupted graphical model has a tree structure is critical to the techniques of this line of work. As Casanellas et al. (2021) astutely observe, when the random vector associated with the true underlying graph is corrupted

with independent but non-identical additive noise, the robust estimation problem reduces to a latent tree structure learning problem. We improve on the algorithmic and theoretical results of this line of work by considering the robust model selection problem for general graphs. Our main contributions are summarized below.

- We establish a fundamental identifiability result (c.f. Theorem 3.1) for general graphs in the robust Gaussian graphical model selection problem. This confirms that the identifiability problem is exacerbated if one considers more general graphs. More importantly, this also generalizes the identifiability results from earlier lines of work and identifies an equivalence class up to which one may hope to recover the underlying structure.
- We devise a novel algorithm, called NoMAD (for Noisy Model selection based on Ancestor Discovery), that tackles the robust model selection problem for *general* graphs extending the results of (Katiyar et al., 2019; 2020; Casanellas et al., 2021; Tandon et al., 2021). Our algorithm is based on a novel "ancestor discovery" procedure (see Section 3.4) that we expect to be of independent interest. It is worth observing that the tree-based algorithms previously proposed fail, often catastrophically, when there are loops in the underlying graph.
- We show that NoMAD provably recovers the underlying graph up to a small equivalence class (c.f. Theorem 4.2) and establish sample complexity results (c.f. Theorem 5.3) for partial structure recovery in the high-dimensional regime.
- We also show the efficacy of our algorithm through experiments on synthetic and realistic network structures.

## 2 Related Work

Several lines of research have tackled the problem of robust estimation of high-dimensional graphical models under corruption. This includes graphical modeling with missing data, outliers, or bounded noise, see, for instance, Loh & Wainwright (2011); Chen et al. (2013); Wang et al. (2014); Nguyen et al. (2022) and references therein. For the missing data problem, several other algorithms have been proposed for estimating mean values and covariance matrices from the incomplete dataset available to the learner RJa & Rubin (1987); Schneider (2001); Lounici (2014). Zheng & Allen (2022) considered a variant of the missing value problem where instead of missing values, the measurements are irregular; that is, different vertex pairs have vastly different sample sizes. Vinci et al. (2019); Chang et al. (2023); Dasarathy (2019) explored the situations where one is only able to obtain samples from subsets of variables, possibly missing joint observations from several pairs. Sun & Li (2012) and Yang & Lozano (2015) proposed algorithms for handling the outliers. There is another line of work that treats this problem using the error-in-variables lens (see the books and papers Hwang (1986); Carroll et al. (1995); Iturria et al. (1999); Xu & You (2007) and references therein). For the problem of model selection from bounded noisy measurements, see Wang et al. (2014); Öllerer & Croux (2015); Loh & Tan (2018); Chen et al. (2015).

However, these papers do not consider the setting of unknown additive noise and the corresponding implications on the conditional independence structure. Recently, Nikolakakis et al. (2019) considered recovering forest-structured graphical models assuming that noise distribution across all vertices is identical. In contrast, our setting allows for unknown and non-identical noise. The robust model selection problem, as considered here, had not been adequately addressed even for the tree-structured graphical models until the recent work by Katiyar et al. (2019; 2020) who showed that the structure recovery in the presence of unknown noise is possible only up to an equivalence class. These studies also proposed algorithms to recover the correct equivalence class from noisy samples. Using information-theoretic methods, Tandon et al. (2021) improved the sample complexity result of Katiyar et al. (2020); Nikolakakis et al. (2019) and provided a more statistically robust algorithm for partial tree recovery. Finally, Zhang & Tan (2021) studied the structure recovery problem under noise when the nodes of the GGM are vector-valued random variables. However, these results are limited to tree-structured graphical models, as they inherently use the additive distance metric property to learn the full (or partial) structure. However, the additive distance metric does not hold for general structures (see more on Section 3.3). The results of this paper significantly extend this line of work, and are applicable to general graphs.

## 3 Preliminaries and Problem Statement

**Graph theory.** Let $G = (V, E)$ be an undirected graph on vertex set $V$ (with cardinality $p$) and edge set $E \subset \binom{V}{2}$. For a vertex $v \in V$, let $N_v \triangleq \{u \in V : \{u, v\} \in E\}$ be the neighborhood of the vertex $v \in V$ and *degree* $\deg(v)$ be the size of $N_v$. A vertex $v$ is said to be *leaf* if $\deg(v) = 1$. A *subgraph* of $G$ is any graph whose vertices and edges are subsets of those of $G$. For $V' \subseteq V$ the *induced subgraph* $G(V')$ has the vertex set $V'$ and the edge set $E' = \{\{u, v\} \in E : u, v \in V'\}$. A *path* between the vertices $u, v$ is a sequence of distinct vertices $v_1 = u, v_2, \ldots, v_k = v$ such that $\{v_i, v_{i+1}\} \in E$, for $1 \leq i < k$. We let $\mathcal{P}_{uv}$ denote the set of all paths between $u$ and $v$. If $\mathcal{P}_{uv}$ is not empty, we say $u$ and $v$ are connected. The graph $G$ is connected if every pair of vertices in $G$ is connected. A set $S \subseteq V$ separates two disjoint subsets $A, B \subseteq V$ if any path from $A$ to $B$ contains a vertex in $S$. We denote this separation as $A \perp\!\!\!\perp B \mid S$. The resemblance of this notation to that of the conditional independence of random variables in the graphical model will be made clear later.

**Gaussian graphical models.** Let $\mathbf{X} = (X_1, X_2, \ldots, X_p) \in \mathbb{R}^p$ be a zero-mean Gaussian random vector with a covariance matrix $\Sigma \in \mathbb{R}^{p \times p}$. Compactly, $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \Sigma)$, where $\mathbf{0}$ is the $p$-dimensional vector of all zeros. Let $G = ([p], E)$ be a graph on the vertex set $[p] \triangleq \{1, 2, \ldots, p\}$ representing the coordinates of $\mathbf{X}$. Let $K \triangleq \Sigma^{-1}$ is called the *precision matrix* of $\mathbf{X}$. The distribution of $\mathbf{X}$ is said to be a *Gaussian graphical model* (or equivalently, Markov) with respect to $G$ if $K_{ij} = 0$ for all $\{i, j\} \notin E$ [1]. In other words, for any $\{i, j\} \notin E$, $X_i$ and $X_j$ are conditionally independent given all the other coordinates of $\mathbf{X}$ (see Lauritzen (1996) for more details). In the sequel, we will use a generic set $V$ to denote the vertex set of our graph with the understanding that every element in $V$ is uniquely mapped to a coordinate of the corresponding random vector $\mathbf{X}$. For a vertex $v \in V$, with a slight abuse of notation, we write $X_v$ to denote the corresponding coordinate of $\mathbf{X}$. Similarly, we write $\Sigma_{uv}$ to mean the covariance between $X_u$ and $X_v$.

### 3.1 The Robust Model Selection Problem

In this paper, we consider a variant of the model selection problem, which we refer to as *robust model selection*. Formally, let $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ be a Gaussian graphical model with respect to an unknown graph $G = (V, E)$. In the robust model selection problem, the goal is to estimate the edge set $E$ (or equivalently the sparsity pattern of $K = \Sigma^{-1}$) when one only has access to noisy samples of $\mathbf{X}$. That is, we suppose we have access to the corrupted version $Y$ of the underlying random vector $X$ such that $\mathbf{Y} = \mathbf{X} + \mathbf{Z}$, where the noise $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, D)$ is independent of $\mathbf{X}$ and $D$ is assumed to be diagonal with possibly distinct and even zero entries. In other words, the noise is assumed to be independent and heteroscedastic while potentially allowing for some coordinates of $\mathbf{X}$ to be observed uncorrupted. Observe that $\mathbf{Y} \sim \mathcal{N}(\mathbf{0}, \Sigma^o)$, where $\Sigma^o \triangleq \Sigma + D$. Indeed, $D$ is assumed to be unknown.

Unfortunately, such corruption can completely obliterate the conditional independence structure of $\mathbf{X}$. For instance, suppose that $D = e_j e_j^T$, where $e_j$ is a vector of zeros except in the $j^{th}$ entry where it is one. By the Sherman-Morrison identity (see e.g., Horn & Johnson, 2012), we have $(\Sigma^o)^{-1} = K - cKe_je_j^\top K$ for some $c \geq 0$. The term $Ke_je_j^\top K$ can be dense in general and, hence, can fully distort the sparsity of $K$ (the conditional independence structure of $\mathbf{X}$).

In view of the above example, the robust model selection problem appears intractable. As outlined in Section 1, recent studies show that this problem is partially tractable for tree-structured graphs. Notably, as Casanellas et al. (2021) astutely observes, one can reduce the problem of robust structure estimation of trees to the problem of learning the structure of latent tree graphical models, which enjoy several efficient algorithms and rich theoretical results (see, e.g., Choi et al. (2011); Erdös et al. (1999); Semple et al. (2003); Dasarathy et al. (2014)). To see this, suppose that $\mathbf{X}$ is Markov according to a graph $G = (V = [p], E)$ that is tree-structured. Let the *joint graph* $G^{\text{J}}$ denote the graph obtained by creating a copy of each node in $G$ and linking the copies to their counterparts in $G$. Formally, we define $G^{\text{J}} = (V^{\text{J}}, E^{\text{J}})$, where $V^{\text{J}} \triangleq V \cup \{1^e, 2^e, \ldots, p^e\}$ and $E^{\text{J}} \triangleq E \cup \{\{i, i^e\} : i \in [p]\}$. In subsequent sections, for any vertex subset $B \subseteq 2^V$, we let $B^e$ denote the vertices associated with the noisy observations from $B$ and call it the *noisy counterpart* of $B$.

[1] Hence, the sparsity pattern of $K$ is represented by the edge set of $G$

Notice that with this definition, if we associate the coordinates of $\mathbf{Y}$ to the newly added leaf vertices, the concatenated random vector $[\mathbf{X}; \mathbf{Y}]$, obtained by stacking $\mathbf{X}$ on top of $\mathbf{Y}$, is Markov according to $G^{\text{J}}$. Casanellas et al. (2021) then uses the fact that when given samples of $\mathbf{Y}$, one can reconstruct a reduced latent tree representation of $G^{\text{J}}$, which in turn can be used to infer an equivalence class of trees that contains the true tree $G$. Indeed, the equivalence class thus obtained is the same one identified by Katiyar et al. (2019; 2020). We next state our general identifiability result after introducing some more graph-theoretic concepts.

### 3.2 An Identifiability Result

A connected graph $G$ is said to be *biconnected* if at least 2 vertices need to be removed to disconnect the graph. A subgraph $H$ of $G$ is said to be a *biconnected component* if it is a maximal biconnected subgraph of $G$. That is, $H$ is not a strict subgraph of any other biconnected subgraph of $G$. The vertex set of such a biconnected component will be referred to as a *block*. A block is *non-trivial* if it has more than two vertices. For example, in Fig. 1a, the vertex set $\{1, 2, 3, 4\}$ and $B_1 \cup \{6, 8\}$ are non-trivial blocks where $B_1$ is an arbitrary set of vertices such that the subgraph on $B_1 \cup \{6, 8\}$ is a biconnected component, whereas, the set $\{10, 8\}$ is a trivial block. In what follows, we will often be interested in the vertices of such non-trivial blocks and toward this we write $\mathcal{B}^{\text{NT}}$ to denote the set of all vertex sets of non-trivial blocks in $G$. From these definitions, it follows that trees (which are cycle free) do not have any non-trivial blocks. It also follows that two blocks can share at most one vertex; we refer to such shared vertices as *cut* vertices. In Fig. 1a, the vertices 4 and 10 are cut vertices. The vertices in $\mathcal{B}^{\text{NT}}$ which are not cut vertices are referred to as *non-cut* vertices. In Fig. 1a, the vertex 1 is a non-cut vertex.

With these definitions, we now introduce a novel representation for a graph $G$ that will be crucial to stating our results. This representation is a tree-structured graph $\mathcal{T}_{\text{AST}}(G)$, which we call the *articulated set tree* of $G$, whose vertices correspond to (a) non-trivial blocks in $G$, and (b) vertices in $G$ that are not a member of any non-trivial blocks. Vertices in this tree-structured representation are connected by edges if the corresponding sets in $G$ either share a vertex or are connected by a single edge. The vertices in the original graph that are responsible for the edges in the representation are called articulation points[2]. We formally define this below. For example, for $G$ in Fig. 1a, the articulated set tree is illustrated in Fig. 1c, where the sets $\{1, 2, 3, 4\}$ and $\{17, 18, 19\}$ are associated to vertices in the articulated set tree representation, and $4, 6, 7, 14$, and $17$ are examples of articulation points.

**Definition 3.1** (Articulated Set Tree)**.** *For an undirected graph $G = (V, E)$, the* **articulated set tree** *$\mathcal{T}_{AST}(G)$ is a tuple $(\mathcal{P}, \mathcal{E}, \mathcal{A})$ where (a) the set $\mathcal{P} = \{B : B \in \mathcal{B}^{NT}\} \cup \{\{v\} : v \in V \setminus \cup_{B \in \mathcal{B}^{NT}} B\}$; (b) an edge $\{P, P'\} \in \mathcal{E}$ if and only if (i) vertices $v, v' \in V$ are such that $v \in P, v' \in P'$, and $\{v, v'\} \in E$ or (ii) there exists a vertex $v \in V$ such that $v \in P \cap P'$, and (c) the articulation function $\mathcal{A} : \mathcal{E} \to V \times V$ returns the articulation points of each edge.*

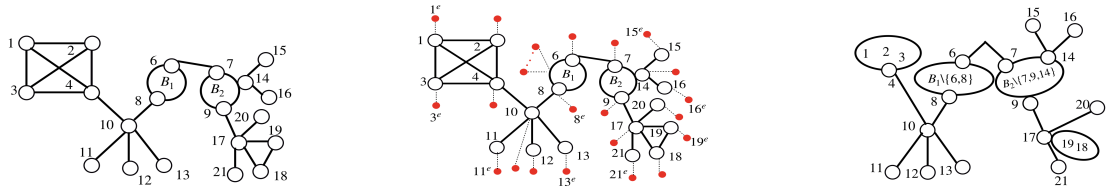
Notice that the articulated set tree (AST), as the name suggests, is indeed a tree. Otherwise, by definition, the set of non-trivial blocks $\mathcal{B}^{\text{NT}}$ would be incorrect (we show this formally in Lemma B.1 in the Appendix). Readers familiar with graph theory may have observed that the AST representation is quite similar to the block-cut tree representation (see e.g., Harary, 1971; Biggs et al., 1986), but unlike a block-cut tree, the subgraph associated with any non-trivial block does not matter in the articulated set tree.

We will now define the equivalence class of graphs up to which robust recovery is possible. Let $L(G)$ denote the set of all leaves in $G$ (i.e., all vertices of degree one). A subset $R \subset L(G)$ is said to be *remote* if no two elements of $R$ share a common neighbor. Let $\mathcal{R}$ be the set of all remote subsets of $L(G)$. For each $R \in \mathcal{R}$, define a graph $G_R$ on $V$ by exchanging each vertex in $R$ with its (unique) neighbor.

**Definition 3.2** (Equivalence Relation, $\sim$)**.** *Two graphs $G, H$ are said to be equivalent if and only if $\exists R \in \mathcal{R}$ such that $\mathcal{T}_{AST}(G_R) = \mathcal{T}_{AST}(H)$. Symbolically, we write as $H \sim G$.*

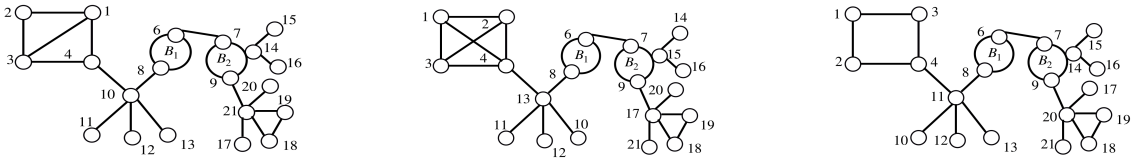
We let $[G]$ denote the equivalence class of $G$ with respect to $\sim$. It is not hard to verify that Definition 3.2 is a valid equivalence relation. Furthermore, it can be readily checked that this notion of equivalence subsumes the ones defined for trees in Katiyar et al. (2019); Casanellas et al. (2021). Fig. 2 illustrates three graphs from

[2]Articulation (points) vertices are cut vertices that separate non-trivial blocks from the rest of the graph.



(a) $G$ (b) $G^J$ (c) $\mathcal{T}_{\text{AST}}(G)$

Figure 1: (a) a true underlying graph where both $B_1 \cup \{6, 8\}$ ($B_2 \cup \{7, 9\}$) are non-trivial blocks where $B_1$ ($B_2$) is an arbitrary set of vertices such that the subgraph on $B_1 \cup \{6, 8\}$ ($B_2 \cup \{7, 9\}$) is a biconnected component, (b) joint graph $G^J$; noisy vertices associated with the non-trivial blocks containing $B_1$ and $B_2$, and some other vertices are not numbered to reduce the clutter, and (c) the articulated set tree $\mathcal{T}_{\text{AST}}(G)$.



(a) $G$ (b) $G_1$ (c) $G_2$

Figure 2: An illustration of three graphs from the same equivalence class of $G$ in Fig. 1a. For all three graphs, $B_1 \cup \{6, 8\}$ ($B_2 \cup \{7, 9\}$) can have any induced subgraph as long as subgraphs on $B_1 \cup \{6, 8\}$ ($B_2 \cup \{7, 9\}$) is a biconnected component.

the same equivalence class. Notice that $G_1$ can be constructed from $G$ by: (i) exchanging the labels between the leaf vertices $\{13, 17, 15\}$ with their corresponding neighbors $\{10, 21, 14\}$; (ii) adding an edge $\{2, 4\}$ inside a non-trivial block. Similarly, $G_2$ can be constructed from $G$ by: (i) exchanging the labels between the leaf vertices $\{11, 20\}$ with their corresponding neighbors $\{10, 21\}$; (ii) removing an edge $\{1, 4\}$ from a non-trivial block[3]. Therefore, for any two graphs in the equivalence class, the non-cut vertices of any non-trivial block remain unchanged, whereas, the edges in the non-trivial block can be arbitrarily changed; the labels of the leaves can be swapped with their neighbor. In the following we will show that our identifiability result also complements the equivalence class. Finally, notice that with the similar operation described above, given a $\mathcal{T}_{\text{AST}}$ which equals to $\mathcal{T}_{\text{AST}}(G)$, one can recover all the graphs in $[G]$.

We now state our unidentifiability result, which establishes that the true covariance matrix of any graph in the equivalence class $[G]$, under the noise model of Subsection 3.1, will result in the same observed covariance matrix $\Sigma^o$.

**Theorem 3.1** (Undentifiability)**.** *Fix a covariance matrix $\Sigma^*$ whose conditional independence structure is given by the graph $G$. Suppose we are given a noisy covariance matrix $\Sigma^o = \Sigma^* + D$ where $D = diag(D_{11}, \ldots, D_{pp}) \geq 0$. Then, for any $H \in [G]$ where $H \neq G$, there exists matrices $\Sigma_H, D_H$ such that $\Sigma^o = \Sigma_H + D_H$, $D_H$ is a diagonal matrix with non-negative entries, and the sparsity pattern of $(\Sigma_H)^{-1}$ is described by $H$.*

This theorem is proved in Section D. Notice that this undentifiability result shows it is impossible to uniquely recover $G$ as there are other confounding graphs whose noisy observations would be indistinguishable from those of $G$. However, this theorem does not rule out the possibility of recovering the equivalence class to which $G$ belongs since all the confounding examples are confined to $[G]$. In Section 3.3, we devise an algorithm that precisely does this. Before we conclude this section, we introduce the notion of partial structure recovery and discuss how recovering the equivalence class still reveals useful information about the true graph.

[3]Notice that the sets $\{13, 17, 15\}$ and $\{11, 20\}$ are remote according to Definition 3.2.

**Partial Structure Recovery of $G$.** Given the noise model described in Subsection 3.1 we know that any graph is only identifiable upto the equivalence relation in Definition 3.2. This is not only the best one can do (by Theorem 3.1), but also preserves useful *partial* structure of the graph. In particular, such a partial structure recovery is able to identify the (non-cut) constituents of the non-trivial blocks and the set of leaf vertices (and neighbors thereof). As the following examples illustrate, we conclude this subsection by arguing that **even such partially recovered graphs** are instrumental in several application domains.

1. **Electrical distribution networks** usually have radial (or globally tree-like) network structures. Recently, several parts of such networks have become increasingly locally interconnected due to the adoption of technologies like roof-top solar panels and battery power storage that enable more flexible power flow. Nonetheless, practitioners critically rely on (learning) the global structure for most operations and maintenance tasks, such as state estimation, power flow, and cybersecurity.

2. **Neuronal networks.** Network models are commonly used to describe structural and functional connectivity in the brain Sporns (2018). An important property of networks that model the brain is their modular structure; modules or communities correspond to clusters of nodes that are densely connected (and are presumably functionally related). Such modular structure has long been regarded as a hallmark of many complex systems; see Herbert et al. (1962) for more details. Module detection, that is, understanding which nodes belong to which modules can yield important insights into how networks function and also uncover a network's latent community structure.

### 3.3 The Robust Model Selection Algorithm

We now present an algorithm that can recover the partial structure of a graph $G$ for which only noisy samples are available based on the setup from Subsection 3.1. Before we describe our algorithm, we introduce a few more concepts that will play a key role. We start with a well-known fact about the factorization of pairwise correlations for a faithful[4] Gaussian graphical model. First, recall that for two random variables $X_i$ and $X_j$, the *correlation coefficient* is defined as $\rho_{ij} \triangleq \Sigma_{ij}/\sqrt{\Sigma_{ii}\Sigma_{jj}}$.

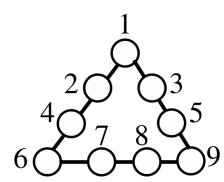



Figure 3: Graph with multiple minimal mutual separators.

**Fact 1** (see e.g., Soh & Tatikonda (2014))**.** *For a faithful Gaussian graphical model, $X_i \perp\!\!\!\perp X_k | X_j$ if and only if $\rho_{ik} = \rho_{ij} \times \rho_{jk}$.*

We now define information distances, which play a vital role in designing our algorithm.

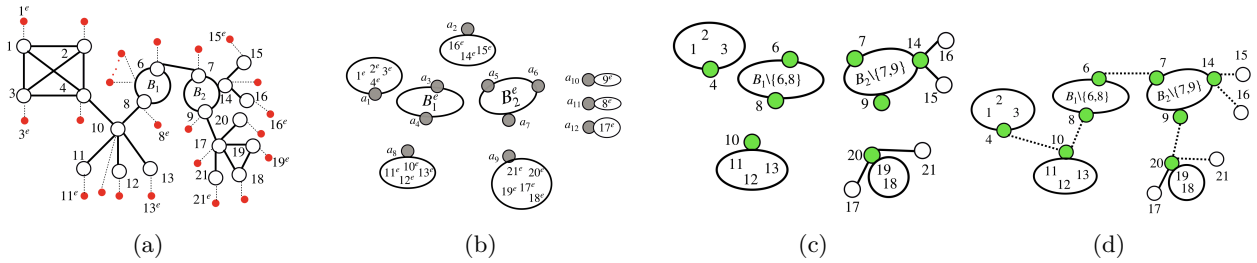
**Definition 3.3** (Information distances)**.** *For $(X_1, \ldots, X_p) \sim \mathcal{N}(\mathbf{0}, \Sigma)$, the information distance between $X_i$ and $X_j$ is defined by $d_{ij} \triangleq -\log|\rho_{ij}| \geq 0$, where $\rho_{ij}$ is the pairwise correlation coefficient between $X_i$ and $X_j$.*

For tree-structured Gaussian graphical models, the strength of the correlation dictates the information (or graphical) distance between vertices $i$ and $j$. Higher the strength, the smaller is the distance, and vice versa. In fact, the information distance defined this way is an additive metric on the vertices of the tree. Although the graphs we consider are not necessarily trees, we still refer to this quantity as a distance throughout the paper for convenience. We now define the notion of ancestors for a triplet of vertices using the notion of minimal mutual separator.

**Definition 3.4** (Minimal mutual separators, Star triplets, Ancestors)**.** *Fix a triplet of vertices $U \in \binom{V}{3}$. A vertex set $S \subseteq V$ is called a mutual separator of $U$ if $S$ separates each pair $i, j \in U$; that is, every path $\pi \in \mathcal{P}_{ij}$ contains at least one element of $S$. The set $S$ is called a* minimal mutual separator *of the triplet $U$ if no proper subset of $S$ is a mutual separator of $U$. We let $S_{\min}(U)$ denote the set of all minimal mutual separators of $U$. $U$ is said to be a* star triplet *if $|S_{\min}(U)| = 1$ and the separator in $S_{\min}(U)$ is a singleton. The unique vertex that mutually separates $U$ is called the* ancestor *of $U$.*

For instance, for the graph in Fig. 3, the set $\{2, 4, 7, 8, 3, 5\}$ is a mutual separator for the triple $\{1, 6, 9\}$. Further notice that minimal mutual separator set may not be unique for a triplet: here, the sets $\{2, 3, 7\}$ and

[4]The global Markov property for GGMs ensures that graph separation implies conditional independence. The reverse implication need not to hold. However, for a faithful GGM, the reverse implication does hold.



(a) (b) (c) (d)

Figure 4: (a) The joint graph $G^{\text{J}}$, (b) the leaf clusters and internal clusters of $G^{\text{J}}$; $B_1^e$ ($B_2^e$) denote the set of noisy vertices associated with the vertices in $B_1$ ($B_2$); Also, recall that clusters are a set of vertices; grey vertices are identified but unlabeled articulation points associated with the clusters, (c) non-trivial blocks and trivial blocks along with the identified and labeled articulation points, and (d) the edges between different articulation points.

$\{4, 5, 8\}$ are minimal mutual separators of the triple $\{1, 6, 9\}$. In Fig. 1a, for the triplet $\{1, 11, 12\}$, minimal mutual separator $S_{\min}(\{1, 11, 12\}) = \{10\}$. Further, notice that an ancestor of $U$ can be one of the elements of $U$. In Fig. 1a, the vertex $\{10\}$ is the ancestor of triplet $\{4, 10, 11\} \triangleq U$. Notice that for triplet $U$, the distance between 10 and the ancestor is zero.

For a star triplet $\{i, j, k\}$ with ancestor $r$, it is clear that the following holds true based on the relationship between graph separation and conditional independence: $X_i \perp\!\!\!\perp X_j \mid X_r$, $X_i \perp\!\!\!\perp X_k \mid X_r$, and $X_j \perp\!\!\!\perp X_k \mid X_r$. As a consequence, from Fact 1 and Definition 3.3, the pairwise distances $d_{ij}, d_{ik},$ and $d_{jk}$ satisfy the following equations: $d_{ij} = d_{ir} + d_{rj}$, $d_{ik} = d_{ir} + d_{rk}$, and $d_{jk} = d_{jr} + d_{rk}$. Some straightforward algebra results in the following identities that allows us to compute the distance between each vertex in $\{i, j, k\}$ and the ancestor vertex $r$. In particular, for any ordering $\{x, y, z\}$ of the set $\{i, j, k\}$ notice that the following is true:

$$d_{xr} = 0.5 \times (d_{xy} + d_{xz} - d_{yz}) \tag{1}$$

For a triplet $U = \{i, j, k\}$, we will let $d_i^U \triangleq \frac{1}{2}(d_{ij} + d_{ik} - d_{jk})$. If $U$ is a star triplet, then $d_i^U$ reveals the distance between $i$ and the ancestor of $U$. However, we do not restrict this definition to star triplets alone. When $U$ is not a star triplet, $d_i^U$ is some arbitrary (operationally non-significant) number; in fact, for non-star triplets this quantity may even be negative. Notice that all vertex triplets in a tree are star triplets. Hence, for a tree-structured graphical model, we can choose any arbitrary triplet and if we can find a vertex $r$ for which $d_i^U = d_{ir}, i \in U$, then we can identify the ancestor of $U$. If such a vertex does not exist, we can deduce the existence of a latent ancestor. Therefore, iterating through all possible triplets, one can recover the true (latent) tree structure underlying the observed variables. In fact, several algorithms in the literature use similar techniques to learn trees (see e.g., Saitou & Nei (1987); Krishnamurthy & Singh (2012); Choi et al. (2011); Dasarathy et al. (2014).

### 3.4 The NoMAD Algorithm

In this section, we introduce our algorithm NoMAD, for Noisy Model selection based on Ancestor Discovery, for robust model selection. We describe NoMAD in the population setting (i.e., in the infinite sample limit) for clarity of presentation; the modifications required for the finite sample setting are discussed in Section 5. In the population setting, NoMAD takes as input the (exact) pairwise information distances $d_{ij}$, for all $i, j$ in the observed vertex set $V^{\text{O}}$, and returns an articulated set tree $\mathcal{T}_{\text{op}} \triangleq (\mathcal{P}_{\text{op}}, A_{\text{op}}, E_{\text{op}})$ (see Definition 3.1). A high level overview of the algorithm is given in Algorithm 1. Its operation may be divided into two main steps: (a) learning $\mathcal{P}_{\text{op}}$ and $A_{\text{op}}$; and (b) learning $E_{\text{op}}$. These steps are summarized in the following. A formal algorithmic listing and a full description can be found in the Appendix.

**(a) Learning $\mathcal{P}_{\text{op}}$ and $A_{\text{op}}$.** Inspired by the aforementioned ancestor based tree reconstruction algorithms, NoMAD first identifies the ancestors[5] in $G^{\text{J}}$, and learns the pairwise distances between them. Notice that

[5]We say that a vertex $r$ is an ancestor if there is a triple $U$ such that $r$ is an ancestor of $U$.

finding the ancestors in $G^{\text{J}}$ is challenging for the following reasons: (a) since $G^{\text{J}}$ is not a tree, some vertex triplets are not star triplets (e.g., $\{1^e, 3^e, 11^e\}$ in Fig. 1b), and (b) a subset of vertices (which may include ancestors) in $G^{\text{J}}$ are unobserved or latent. Hence, we can not guarantee the identification of a star triplet following the procedure for trees. NoMAD instead uses a novel procedure that compares *two triplets* of vertices to identify the ancestors; we call this the TIA (Test Identical Ancestor) test which is defined as follows:

**Algorithm 1** NoMAD

1: **Input:** Pairwise distances $\mathcal{D} = \{d_{ij}\}_{i,j \in V^{\text{O}}}$.
2: **Output:** $\mathcal{T}_{\text{op}} \triangleq (\mathcal{P}_{\text{op}}, A_{\text{op}}, E_{\text{op}})$.
3: IdentifyAncestors(Subroutine 3). **Accepts:** $\mathcal{D}$; **Returns:** (I) A set $A_{\text{obs}}$ ($A_{\text{hid}}$ resp.) of observed (hidden resp.) ancestors, and the corresponding collection of vertex triplets $\mathfrak{V}_{\text{obs}}$ ($\mathfrak{V}_{\text{hid}}$ resp.), and (II) The set of pairwise distances $\{d_{ij}\}$ for each pair $i, j \in V^{\text{O}} \cup A_{\text{hid}}$
4: LearnClusters (Subroutine 4). **Accepts**: $A_{\text{obs}}, A_{\text{hid}}$, and $\mathcal{D}$; **Returns:** A collection of (I) leaf clusters $\mathcal{L}$, and (II) internal clusters $\mathcal{I}$.
5: VertexSet-AST(Subroutine 6).**Accepts:** $\mathcal{L}$ and $\mathcal{I}$; **Returns:** (I) the vertex set $\mathcal{P}_{\text{op}}$ (II) the articulation points $A_{\text{op}}$, and (III) a *subset* of the edge set $E_{\text{op}}$.
6: EdgeSet-AST(Subroutine 8). **Accepts:** $\mathcal{P}_{\text{op}}$ and $A_{\text{op}}$; **Returns:** $E_{\text{op}}$.

**Definition 3.5** (TIA Test)**.** *The Test Identical Ancestor (TIA) test accepts a triplet pair $U, W \in \binom{V^{\text{O}}}{3}$, and returns* True *if and only if for all $x \in U$, there exists at least one pair $y, z \in W$ such that $d_x^U + d_y^W = d_{xy}$ and $d_x^U + d_z^W = d_{xz}$.*

In words, in order for a pair of triplets $U, W$ to share an ancestor in $G^{\text{J}}$, each vertex in one triplet (say, $U$) needs to be separated from at least a pair in $W$ by the (shared) ancestor in $G^{\text{J}}$. We now describe the fist step of the NoMAD in three following sub-steps:

Identifying the ancestors in $G^{\text{J}}$. In the first sub-step, NoMAD (a) uses the TIA test to create the set $\mathfrak{V} = \{\mathcal{V} \subset \binom{V}{3} : \text{each } U \in \mathcal{V} \text{ has the same ancestor }\}$, (b) it then assigns each triplet collection $\mathcal{V} \in \mathfrak{V}$ to either $\mathfrak{V}_{\text{obs}}$ or $\mathfrak{V}_{\text{hid}}$; the former is the collection of vertex triples whose ancestor is observed and the latter has ancestors that are hidden, and (c) identifies the observed ancestors and enrolls them into a set of observed ancestors $A_{\text{obs}}$. Furthermore, for each collection $\mathcal{V}_i \in \mathfrak{V}_{\text{hid}}$, NoMAD introduces a hidden vertex, and enrolls it in $A_{\text{hid}}$ such that, $|A_{\text{hid}}|$ equals to the number of hidden ancestors in $G^{\text{J}}$. For example, consider the joint graph $G^{\text{J}}$ in Fig. 4a. In $G^{\text{J}}$, the vertex $\{4\}$ is the observed ancestor of the pair $\{1^e, 4^e, 10\}$ and $\{3^e, 4^e, 8^e\}$, and the vertex $\{8\}$ is the hidden ancestor of the pair $\{3^e, 8^e, 9^e\}$, $\{1^e, 8^e, 7^e\}$. Complete pseudocode for this step appears in Subroutine 3 in Appendix.

Extending the distance set $\{d_{ij}\}_{i,j \in V^{\text{O}}}$. In the next sub-step, using pairwise distances $\{d_{ij}\}_{i,j \in V^{\text{O}}}$, and $A_{\text{hid}}$, NoMAD learns the following distances: (a) $d_{ij}$ for all $i, j \in A_{\text{hid}}$, and (b) $d_{ij}$ for all $i, j \in A_{\text{hid}} \cup V^{\text{O}}$. For learning (a), notice from the last sub-step that each $a_i \in A_{\text{hid}}$ is assigned to a collection of triplets $\mathcal{V}_i \in \mathfrak{V}_{\text{hid}}$. In order to compute the distance between two hidden ancestors (say $a_p, a_q \in A_{\text{hid}}$), the NoMAD chooses two triplets $V_p \in \mathcal{V}_p$ and $V_q \in \mathcal{V}_q$, and computes the set $\Delta_{pq}$ as follows: $\Delta_{pq} = \left\{ d_{xy} - (d_x^{V_p} + d_y^{V_q}) : x \in V_p, y \in V_q \right\}$. Then, the most frequent element in $\Delta_{pq}$, i.e., mode($\Delta_{pq}$), is declared as $d_{pq}$. We show in Appendix that NoMAD not only correctly learns the distance set $\{d_{ij}\}_{i,j \in A_{\text{hid}}}$, but also learns $d_{ij}$ for any $i \in A_{\text{hid}}$ and $j \in V^{\text{O}}$. A pseudocode for this step appears in Subroutine 3 in Appendix.

Learning $\mathcal{P}_{\text{op}}$ and $A_{\text{op}}$. In the final sub-step, NoMAD learns the clusters of vertices in $V^{\text{O}} \setminus A_{\text{obs}}$ using the separation test in Fact 1 which eventually lead to finding $\mathcal{P}_{\text{op}}$ and $A_{\text{op}}$. Specifically, NoMAD learns (a) all the leaf clusters, each of which is a set of vertices that are separated from the rest of the graph by a single ancestor, and (b) all the internal clusters, each of which is a set of vertices that are separated from the rest of the graph by multiple ancestors. For example, in Fig. 4b, the set $\{17^e, 18^e, 19^e, 20^e, 21^e\}$ is a leaf cluster— separated from the rest of the graph by the (hidden) ancestor $a_9$. The set $B_2^e$ is an internal

cluster— separated from the rest of the graph by the set of ancestors $\{a_5, a_6, a_7\}$. Next, NoMAD uses the clusters to learn $\mathcal{P}_{\text{op}}$ and $A_{\text{op}}$ for $\mathcal{T}_{\text{op}}$ by applying the TIA test on each cluster to identify the non-cut vertices and potential cut vertices in it. For example, for the leaf cluster $\{17^e, 18^e, 19^e, 20^e, 21^e\}$, 18 and 19 are non-cut vertices, and a vertex from $17, 20,$ and 21 may be declared as a cut-vertex arbitrarily (see Fig. 4c). A pseudocode for this step appears as Subroutine 4 and Subroutine 6 in Appendix.

**(b) Learning $E_{\text{op}}$.** In this step, NoMAD learns the edge set $E_{\text{op}}$. Notice from Definition 3.1 that any two elements of $\mathcal{P}_{\text{op}}$ are connected with each other through their respective articulation points. Hence, in order to learn $E_{\text{op}}$, NoMAD needs to learn the neighborhood of each articulation point in $G$. To this end, NoMAD first learns this neighborhood for each articulation point in $G$ using Fact 1. Then, in the next step, NoMAD creates an edge between two elements of $\mathcal{P}_{\text{op}}$ if the articulation points from each element are neighbors in $G$ (dotted lines in Fig. 4d). A pseudocode for this step appears as Subroutine 8 in Appendix.

## 4 Performance Analysis of NoMAD in the Population Setting

In this section, we show the correctness of NoMAD in returning the equivalence class of a graph $G$ while having access only to the noisy samples according to the problem setup in Section 3.1. We now make an assumption that will be crucial to show the correctness of NoMAD. This is similar to the faithfulness assumption common in the graphical modeling literature Choi et al. (2011); Kalisch & Bühlman (2007); Uhler et al. (2013), and like the latter, it rules out "spurious cancellations". To that end, let $\mathcal{V}_{\text{star}} \subseteq \binom{V}{3}$ be the set of all star triplets in $G$ (see Definition 3.4). Let $\mathcal{V}_{\text{sep}} \subseteq \binom{V}{3}$ be the set of triplets $V$ such that one of the vertices in $V$ separates the other two vertices.

**Assumption 4.1** (Ancestor faithfulness)**.** *Let $U, W \in \binom{V}{3} \setminus \mathcal{V}_{\text{star}} \cup \mathcal{V}_{\text{sep}}$. Then, (i) there are no vertices $x \in U$ and $a \in W$ that satisfy $d_x^U + d_a^W = d_{xa}$, and (ii) there does not exist any vertex $r \in V$ and $x \in U$ for which the distance $d_{xr}$ satisfies relation in equation 1.*

Notice that Assumption 4.1 is only violated when there are explicit constraints on the corresponding covariance values which, like the faithfulness assumption, only happens on a set of measure zero. We next state the main result of our paper in the population settings.

**Theorem 4.2.** *Consider a covariance matrix $\Sigma^*$ whose conditional independence structure is given by the graph $G$, and the model satisfies Assumption 4.1. Suppose that according to the problem setup in Section 3.1, we are given pairwise distance $d_{ij}$ of a vertex pair $(i, j)$ in the observed vertex set $V^O$, that is, $d_{ij} \triangleq -\log|\rho_{ij}|$ where $\rho_{ij} \triangleq \Sigma^o_{ij}/\sqrt{\Sigma^o_{ii}\Sigma^o_{jj}}$. Then, given the pairwise distance set $\{d_{ij}\}_{i,j \in V^o}$ as inputs, NoMAD outputs the equivalence class $[G]$.*

**Proof Outline.** In order to show that NoMAD correctly learns the equivalence class, it suffices to show that it can correctly deduce the articulated set tree $\mathcal{T}_{\text{op}}$. Given this, and the equivalence relation from Definition 3.2, the entire equivalence class can be readily generated. Our strategy will be to show that NoMAD learns $\mathcal{T}_{\text{op}}$ correctly by showing that it learns (a) the vertex set $\mathcal{P}_{\text{op}}$, (b) the articulation points $A_{\text{op}}$, and (c) the edge set $E_{\text{op}}$ correctly. We will now establish (a) and (b).

- From the description of the algorithm in Section 3.3, it is clear that NoMAD succeeds in finding the ancestors, which is the first step, provided the TIA tests succeed. Indeed, in the first stage of our proof, we establish Lemma B.7 in Appendix which shows that the TIA test passes with two triplets if and only if they share a common ancestor in $G^{\text{J}}$.
- Next, we show in Lemma B.9 and Claim 3 in Appendix that NoMAD correctly learns the distances $d_{ij}$ for all vertices $i, j$ that either in the set of observed vertices $V^{\text{O}}$ or in the set of hidden ancestors $A_{\text{hid}}$. Proposition B.13 establishes the correctness of NoMAD in learning the leaf clusters and internal clusters, and in learning $\mathcal{P}_{\text{op}}$ and $A_{\text{op}}$. The proof correctness of this step crucially depends on identifying the non-cut vertices in $G$ from different clusters which is proved in Lemma B.12.

Finally, we outline the correctness of NoMAD in learning the edge set $E_{\text{op}}$. Recall that NoMAD learns the neighbor articulation points of each articulation point. Proposition B.14 in Appendix shows that NoMAD

correctly achieves this task. Using the neighbors of different articulation points, NoMAD correctly learns the edges between different elements in $\mathcal{P}_{\text{op}}$.

## 5 Performance Analysis of NoMAD in Finite Sample Setting

In describing NoMAD (cf. Section 3.4) and in the analysis in the population setting (cf. Section 4), we temporarily assumed that we have access to the actual distances $d_{ij}$ for the sake of exposition. However, in practice, these distances need to be estimated from samples $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_n\}$. In what follows, we show that NoMAD, with high probability, correctly outputs the equivalence class $[G]$ even if we replace $d_{ij}$ with the estimate $\widehat{d}_{ij} = -\log\left|\widehat{\Sigma^o}_{ij}/\sqrt{\widehat{\Sigma^o}_{ii}\widehat{\Sigma^o}_{jj}}\right|$, where $\widehat{\Sigma^o}_{ij}$ is the $(i,j)$-th element in $\frac{1}{n}\sum_{i=1}^{N}\mathbf{Y}_i\mathbf{Y}_i^T$. We recall from Section 3.4 that the subroutines in NoMAD depend on the TIA test, which relies on the distances $d_{ij}$. Thus, we establish the correctness of NoMAD in the finite sample setting by showing that the empirical TIA (TIA with estimated distances) correctly identifies the ancestors in $G^J$ with high probability. We begin with the following assumptions.

**Assumption 5.2.** *[$\gamma$-Strong Faithfulness Assumption] For any vertex triplet $i,j,k \in \binom{V^o}{3}$, if $i \not\perp\!\!\!\perp j|k$, then $|d_{ij} - d_{ik} - d_{jk}| > \gamma$.*

$\gamma$-Strong Faithfulness assumption is a standard assumption used in the literature (Kalisch & Bühlman (2007); Uhler et al. (2013)). 0-Strong-Faithfulness is just the usual Faithfulness assumption discussed in Section 3.3. This motivates our next assumption which strengthens our requirement on "spurious cancellations" involving ancestors.

**Assumption 5.3** (Strong Ancestor consistency)**.** *For any triplet pair $U, W \in \binom{V}{3} \setminus \mathcal{V}_{\text{star}} \cup \mathcal{V}_{\text{sep}}$ and any vertex pair $(x,a) \in U \times W$, there exists a constant $\zeta > 0$, such that $\left|d_x^U + d_a^W - d_{xa}\right| > \zeta$.*

Assumption 5.3 is in direct analogy with Assumption 5.2. As we show in Lemma B.7, for any pair $U, W \in \binom{V}{3} \setminus \mathcal{V}_{\text{star}} \cup \mathcal{V}_{\text{sep}}$ (i.e., any pair that would fail the TIA test), there exists at least one triplet $\{x,a,b\}$ where $x \in U$ and $a,b \in W$ such that $d_{xa} - d_x^U - d_a^W \neq 0$ and $d_{xb} - d_x^U - d_b^W \neq 0$. This observation motivates us to replace the exact equality testing in the TIA test in Definition 3.5 with the following hypothesis test against zero: $\max\left\{\left|\widehat{d}_{xa} - \widehat{d}_x^U - \widehat{d}_a^W\right|, \left|\widehat{d}_{xb} - \widehat{d}_x^U - \widehat{d}_b^W\right|\right\} \leq \xi$. We set $\xi < \frac{\zeta}{2}$.

Furthermore, in order to learn the distance between two hidden ancestors in $A_{\text{hid}}$, the mode test introduced in Subsection 3.4 needs to be replaced with a finite sample version; we call this the $\epsilon_d$-mode test and this is formalized in Appendix C. Finally, for any triplet $(i,j,k) \in \binom{V^o}{3}$, in order to check whether $i \perp\!\!\!\perp j|k$, the test in Fact 1 needs to be replaced as follows: $|\widehat{d}_{ij} - \widehat{d}_{ik} - \widehat{d}_{jk}| < \frac{\epsilon_d}{6}$. We now introduce two new notations to state our main result. Let $\rho_{\min}(p) = \min_{i,j \in \binom{p}{2}} |\rho_{ij}|$ and $\kappa(p) = \log((16 + (\rho_{\min}(p))^2\,\epsilon_d^2)/(16 - (\rho_{\min}(p))^2\,\epsilon_d^2))$, where $\epsilon_d = \min(\frac{\xi}{14}, \gamma)$, where $\gamma$ is from Assumption 5.2.

**Theorem 5.3.** *Suppose the underlying graph $G$ of a faithful GGM satisfies Assumptions 5.2-5.3. Fix any $\tau \in (0,1]$. Then, there exists a constant $C > 0$ such that if the number of samples $n$ satisfies $n > C\left(\frac{1}{\kappa(p)}\right)\max\left(\log\left(\frac{p^2}{\tau}\right), \log\left(\frac{1}{\kappa(p)}\right)\right)$, then with probability at least $1-\tau$, NoMAD accepting $\widehat{d}_{ij}$ outputs the equivalence class $[G]$.*

**Remark 5.1.** *Theorem 5.3 indicates that the sample complexity of the NoMAD is dependent on the absolute minimum and maximum pairwise correlations $\rho_{\min}$ and $\rho_{\max}$, the number of vertices $p$, and the magnitude of the quantity from Assumption 5.3. Specifically, in regimes of interest, we see that the sample complexity scales as a logarithm in the number of vertices $p$ and inversely in $\rho_{\min}^2(p)$ and $\epsilon_d^2$, thus allowing for robust model selection in the high-dimensional regime. Further note that our sample requirement can have an exponential dependence on $p$ based on the degree of $G$. However, we expect that this dependency can be improved; see Section 7 for more on this avenue for future work.*

(a) Synthetic graph, $G_{\text{syn}}$

(b) IEEE 33 Bus Graph

Figure 5: Synthetic graph and IEEE-33 Bus system considered for our simulation

## 6 Experiments

We perform experiments on both a synthetic graph and an IEEE 33 bus system, which is a graphical representation of established IEEE-33 bus benchmark distribution system (Baran & Wu, 1989), to assess the validity of our theoretical results and to demonstrate the performance of NoMAD. In particular, our experiments demonstrate how an unmodified graphical modeling algorithm (in this case GLASSO Friedman et al. (2008)) compares to NoMAD. The synthetic graph $G_{\text{syn}}$ we consider and graph associated with the IEEE-33 bus system are given in Fig. 5a and Fig. 5b, respectively. As a first step, we need some new performance metrics to make a fair comparison in light of the unidentifiability of the underlying graph structure from noisy data. In the following we will introduce some sets, and show that if an algorithm can recover all these sets, then the output graph (from that algorithm) is in equivalence class.

1. **Families.** For a vertex $i$, define its *family* $F_i$ as $\{v : \deg(v) = 1 \text{ and } \{v, i\} \in E(G)\} \cup \{i\}$. Notice that the subgraph associated with each family is a tree. For synthetic graph $G_{\text{syn}}$, the sets $\{1, 2, 3\}$, $\{8, 9, 10\}$ are some of the families in $G_{\text{syn}}$; For IEEE 33 Bus Graph $\{15, 16, 17, 18\}$, $\{30, 31, 32, 33\}$ are some of the families. Let $\mathcal{F} = \bigcup_{i \in V} F_i$.

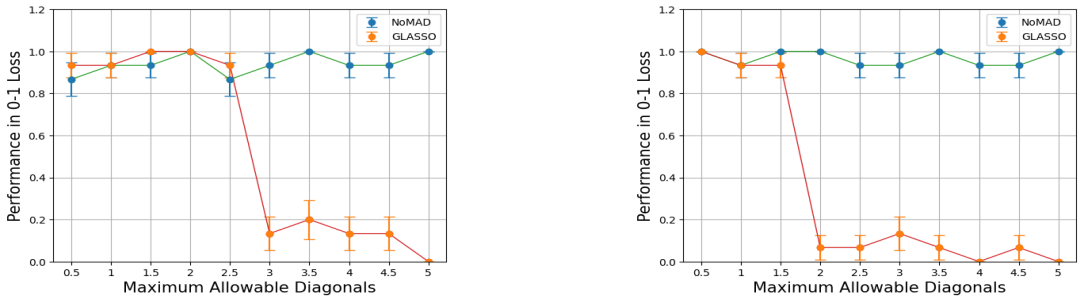
2. **Non-Cut Vertices.** For graph $G$, let $B_{\text{non-cut}}$ be the set of all non-cut vertices in a non-trivial block $B$. Define $\mathcal{B}_{\text{non-cut}} \triangleq \bigcup_{B \in \mathcal{B}^{\text{NT}}} B_{\text{non-cut}}$, where $\mathcal{B}^{\text{NT}}$ is the set of all non-trivial blocks. Recall from Section 3 that in all the equivalent graphs the set of non-cut vertices remain unchanged.

3. **Global Edges.** Let $K$ be the set of cut vertices who do not share an edge with a leaf vertex in $G$. For any vertex $k \in K$, let a family $F_k \in \mathcal{F}$ be such that there exists a vertex $f \in F_k$ such that $\{k, f\} \in E(G)$. Now, we will note two following condition for any neighbor $i \in N(k)$: (a) if $i \in K$, then $\{i, k\} \in E(G)$, and (b) otherwise, there exists a vertex $j \in F_k$ such that $\{j, k\} \in E(G)$. If condition (a) and condition (b) are met, global edge associated with $i$ are recovered correctly .

In Lemma B.15, we demonstrate that if an algorithm learns these sets correctly, and the conditions associated with the global edges are met, then an equivalent graph can be recovered.

**Experimental Setup**. We now describe our experimental setup. We generated precision matrices associated with $G_{\text{syn}}$, and for IEEE-33 bus system, we added some loops. We then inverted the corresponding precision matrices to obtain the respective covariance matrices in the population setting. Next, we added a diagonal matrix of random positive values to the population covariance matrices at each entry to generate the corresponding noisy covariance matrices. Our code is available at https://github.com/ZahinAbrar/NoMAD/blob/main/NoMAD.

We will now compare the performance of 0-1 loss of NoMAD to GLASSO under the influence of noise. In order to compare, our goal is to compare NoMAD with the *best* GLASSO. Our protocol for selecting the *best* GLASSO is as follows: for a fixed maximum allowable diagonals, we selected the regularization parameter for which the output graph given the noise covariance matrix to GLASSO is in equivalence class. Then, for that fixed regularization parameter we report the performance of GLASSO for varying numbers of (increased)




(a) Performance for $G_{\text{syn}}$

(b) Performance for IEEE-33 Bus.

Figure 6: Performance of GLASSO and NoMAD for various degree of noises. Up to the values of 2.5 and 1.5 dollars for $G_{\text{syn}}$ and IEEE-33 Bus, respectively, GLASSO successfully reconstructed an equivalent graph. To investigate this influence, we conducted the experiment with a fixed sample size across 15 trials. In contrast, NoMAD consistently demonstrated its ability to recover an equivalent graph under various noise influences.

maximum allowable diagonals. Notice that the maximum allowed values of the diagonal elements of the matrix $D$ (which is being added to the generated covariance matrix) contain information about the potential influence of the noise in the setup. In order to study this influence we ran the experiment for a fixed sample size over 15 trials. In Fig. 6, we observe that up to the value of 2.5 and 1.5 for $G_{\text{syn}}$ and IEEE-33 Bus, respectively, GLASSO was able to recover an equivalent graph. On the contrary, NoMAD was able to show a consistent performance in recovering an equivalent graph with various influence of noises.

## 7 Conclusion and Future Directions

**Conclusion.** In this paper, we consider model selection of non tree-structured Gaussian graphical models when the observations are corrupted by independent but non-identically distributed noise with unknown statistics. We first show that this ambiguity is unavoidable. Finally, we devise a novel algorithm, referred to as NoMAD, which learns structure up to a small equivalence class.

**Future Directions.** This paper opens up several exciting avenues for future research. First, our novel ancestor testing method can be used to identify the ancestors for other graphical models (beyond Gaussians) where information distance satisfies the factorization property in Fact 1; e.g., Discrete graphical models, where the random vector $\mathbf{X}$ takes values in the product space $\mathcal{X}^p$, where $|\mathcal{X}| = k$. For any $i, j \in [p]$, let $\Upsilon_{ij} \in \mathbb{R}^{k \times k}$ denote the tabular representation of the marginal distribution of the pair $(X_i, X_j)$ and $\Upsilon_{ii}$ denote a diagonal matrix with the marginal distribution of $X_i$ on the diagonal. Then, it is known that the following quantity: $d_{ij} = \frac{\det(\upsilon_{ij})}{\sqrt{\det(\upsilon_{ii}\upsilon_{jj})}}$, can be taken to be the information distance. We refer the reader to Semple et al. (2003) for more on this, including a proof of an equivalent version of Fact 1. Note that the Ising model is a special case, and our results naturally extend to them.

Second, it is well known that $\rho_{\min}$ could scale exponentially in the diameter of the graph; this could imply that the sample complexity will scale polynomially in the number of vertices $p$ even for balanced binary trees, and as bad as exponential for more unbalanced graphs. Now, notice that in Subroutine 3, NoMAD identifies all the star triplets for any ancestor in $G^{\text{J}}$. Hence, this identification procedure in quite computationally expensive. As we reason this in theoretical section of the appendix that this computation is required in order to learn the pairwise distances $d_{ij}$ for each pair $(i, j)$ such that $i \in V^{\text{O}}$ and $j \in A_{\text{hid}}$, where $V^{\text{O}}$ and $A_{\text{hid}}$ is the set of observed vertices, and hidden ancestors, respectively. A promising future research work is to develop a TIA test which can obtain $\{d_{ij}\}_{i,j \in V^{\text{O}} \cup A_{\text{hid}}}$ without iterating all the triplets in $\binom{V^{\text{O}}}{3}$. Furthermore, the Subroutine 4 can be redesigned to a computationally efficient one by learning the clusters in *divide and conquer* manner. Another promising avenue for future research work is to obtain an upper bound on the diagonal entries $D_{ii}$ for which the underlying graph $G$ is identifiable. Finally, future research can be done to

understand to understand the behavior of the hyperparameters $\gamma$ and $\zeta$ using the stability selection method (see Meinshausen & Bühlmann, 2010) in finite sample settings.

## References


Rajasekhar Anguluri, Gautam Dasarathy, Oliver Kosut, and Lalitha Sankar. Grid topology identification with hidden nodes via structured norm minimization. *IEEE Control Systems Letters*, 6:1244–1249, 2022.

Mesut E Baran and Felix F Wu. Network reconfiguration in distribution systems for loss reduction and load balancing. *IEEE Transactions on Power delivery*, 4(2):1401–1407, 1989.

Norman Biggs, E Keith Lloyd, and Robin J Wilson. *Graph Theory*. Oxford University Press, 1986.

Edward T Bullmore and Danielle S Bassett. Brain graphs: graphical models of the human brain connectome. *Annual review of clinical psychology*, 7:113–140, 2011.

Raymond J Carroll, David Ruppert, and Leonard A Stefanski. *Measurement error in nonlinear models*, volume 105. CRC press, 1995.

Marta Casanellas, Marina Garrote-López, and Piotr Zwiernik. Robust estimation of tree structured models. *arXiv preprint arXiv:2102.05472*, 2021.

Andersen Chang, Lili Zheng, Gautam Dasarathy, and Genevera I Allen. Nonparanormal graph quilting with applications to calcium imaging. *Stat*, 12(1):e623, 2023.

Mengjie Chen, Chao Gao, and Zhao Ren. Robust covariance matrix estimation via matrix depth. *arXiv preprint arXiv:1506.00691*, 2015.

Yudong Chen, Constantine Caramanis, and Shie Mannor. Robust sparse regression under adversarial corruption. In *International Conference on Machine Learning*, pp. 774–782. PMLR, 2013.

Myung Jin Choi, Vincent YF Tan, Animashree Anandkumar, and Alan S Willsky. Learning latent tree graphical models. *J. of Machine Learning Research*, 12:1771–1812, 2011.

Gautam Dasarathy. Gaussian graphical model selection from size constrained measurements. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 1302–1306. IEEE, 2019.

Gautam Dasarathy, Robert Nowak, and Sebastien Roch. Data requirement for phylogenetic inference from multiple loci: a new distance method. *IEEE/ACM transactions on computational biology and bioinformatics*, 12(2):422–432, 2014.

Gautam Dasarathy, Elchanan Mossel, Robert Nowak, and Sebastien Roch. A stochastic farris transform for genetic data under the multispecies coalescent with applications to data requirements. *Journal of Mathematical Biology*, 84(5):1–37, 2022.

Deepjyoti Deka, Ross Baldick, and Sriram Vishwanath. One breaker is enough: Hidden topology attacks on power grids. In *2015 IEEE Power & Energy Society General Meeting*, pp. 1–5. IEEE, 2015.

Deepjyoti Deka, Saurav Talukdar, Michael Chertkov, and Murti V Salapaka. Graphical models in meshed distribution grids: Topology estimation, change detection & limitations. *IEEE Transactions on Smart Grid*, 11(5):4299–4310, 2020.

Mathias Drton and Marloes H Maathuis. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4:365–393, 2017.

Péter L Erdös, Michael A Steel, LászlóA Székely, and Tandy J Warnow. A few logs suffice to build (almost) all trees: Part ii. *Theoretical Computer Science*, 221(1-2):77–118, 1999.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

F. Harary. *Graph Theory*. Addison Wesley series in mathematics. Addison-Wesley, 1971.

Simon Herbert et al. The architecture of complexity. *Proceedings of the American Philosophical Society*, 106 (6):467–482, 1962.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Jiunn T Hwang. Multiplicative errors-in-variables models with applications to recent data released by the us department of energy. *Journal of the American Statistical Association*, 81(395):680–688, 1986.

Stephen J Iturria, Raymond J Carroll, and David Firth. Polynomial regression and estimating functions in the presence of multiplicative measurement error. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):547–561, 1999.

Markus Kalisch and Peter Bühlman. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8(3), 2007.

Ashish Katiyar, Jessica Hoffmann, and Constantine Caramanis. Robust estimation of tree structured gaussian graphical models. In *International Conference on Machine Learning*, pp. 3292–3300. PMLR, 2019.

Ashish Katiyar, Vatsal Shah, and Constantine Caramanis. Robust estimation of tree structured ising models. *arXiv preprint arXiv:2006.05601*, 2020.

Minje Kim and Paris Smaragdis. Single channel source separation using smooth nonnegative matrix factorization with markov random fields. In *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2013.

Akshay Krishnamurthy and Aarti Singh. Robust multi-source network tomography using selective probes. In *2012 Proceedings IEEE INFOCOM*, pp. 1629–1637. IEEE, 2012.

Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

Po-Ling Loh and Xin Lu Tan. High-dimensional robust precision matrix estimation: Cellwise corruption under *epsilon*-contamination. *Electronic Journal of Statistics*, 12(1):1429–1467, 2018.

Po-Ling Loh and Martin J Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. *Advances in Neural Information Processing Systems*, 24, 2011.

Karim Lounici. High-dimensional covariance matrix estimation with missing observations. *Bernoulli*, 20(3): 1029–1058, 2014.

Marloes Maathuis, Mathias Drton, Steffen Lauritzen, and Martin Wainwright. *Handbook of graphical models*. CRC Press, 2018.

Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 72(4):417–473, 2010.

Kevin Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. *arXiv preprint arXiv:1301.6725*, 2013.

Viet Anh Nguyen, Daniel Kuhn, and Peyman Mohajerin Esfahani. Distributionally robust inverse covariance estimation: The wasserstein shrinkage estimator. *Operations Research*, 70(1):490–515, 2022.

Konstantinos E Nikolakakis, Dionysios S Kalogerias, and Anand D Sarwate. Learning tree structures from noisy data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1771–1782. PMLR, 2019.

Viktoria Öllerer and Christophe Croux. Robust high-dimensional precision matrix estimation. In *Modern nonparametric, robust and multivariate methods*, pp. 325–350. Springer, 2015.

Thomas Ott and Ruedi Stoop. The neurodynamics of belief propagation on binary markov random fields. *Advances in neural information processing systems*, 19, 2006.

Little RJa and DB Rubin. *Statistical analysis with missing data*. 1987.

Naruya Saitou and Masatoshi Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4):406–425, 1987.

Tapio Schneider. Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values. *Journal of climate*, 14(5):853–871, 2001.

Charles Semple, Mike Steel, et al. *Phylogenetics*, volume 24. Oxford University Press on Demand, 2003.

De Wen Soh and Sekhar C Tatikonda. Testing unfaithful gaussian graphical models. *Advances in Neural Information Processing Systems*, 27:2681–2689, 2014.

Olaf Sporns. Graph theory methods: applications in brain networks. *Dialogues in clinical neuroscience*, 2018.

Hokeun Sun and Hongzhe Li. Robust gaussian graphical modeling via l1 penalization. *Biometrics*, 68(4): 1197–1206, 2012.

Anshoo Tandon, Aldric HJ Yuan, and Vincent YF Tan. SGA: A robust algorithm for partial recovery of tree-structured graphical models with noisy samples. *arXiv preprint arXiv:2101.08917*, 2021.

Caroline Uhler, Garvesh Raskutti, Peter Bühlmann, and Bin Yu. Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics*, pp. 436–463, 2013.

Giuseppe Vinci, Gautam Dasarathy, and Genevera I Allen. Graph quilting: graphical model selection from partially observed covariances. *arXiv preprint arXiv:1912.05573*, 2019.

Jun-Kun Wang et al. Robust inverse covariance estimation under noisy measurements. In *International Conference on Machine Learning*, pp. 928–936. PMLR, 2014.

Qinfeng Xu and Jinhong You. Covariate selection for linear errors-in-variables regression models. *Communications in Statistics—Theory and Methods*, 36(2):375–386, 2007.

Eunho Yang and Aurélie C Lozano. Robust gaussian graphical modeling with the trimmed graphical lasso. *Advances in Neural Information Processing Systems*, 28, 2015.

Fengzhuo Zhang and Vincent Tan. Robustifying algorithms of learning latent trees with vector variables. *Advances in Neural Information Processing Systems*, 34, 2021.

Lili Zheng and Genevera I Allen. Graphical model inference with erosely measured data. *arXiv preprint arXiv:2210.11625*, 2022.

Yiming Zuo, Yi Cui, Guoqiang Yu, Ruijiang Li, and Habtom W Ressom. Incorporating prior biological knowledge for network-based differential gene expression analysis using differentially weighted graphical lasso. *BMC bioinformatics*, 18(1):1–14, 2017.

# Appendix
# Robust Model Selection of Gaussian Graphical Models

## A Algorithmic Details

In the population setting, NoMAD takes as input the pairwise distances $d_{ij}$, for all $i, j$ in the observed vertex set $V^{\mathrm{O}}$, and returns an articulated set tree $\mathcal{T}_{\mathrm{op}} \triangleq (\mathcal{P}_{\mathrm{op}}, A_{\mathrm{op}}, E_{\mathrm{op}})$ (see Definition 3.1). Its operation is divided into two main steps: (a) learning $\mathcal{P}_{\mathrm{op}}$ and $A_{\mathrm{op}}$; and (b) learning $E_{\mathrm{op}}$. These steps are summarized in the following.

### A.1 Learning $\mathcal{P}_{\mathrm{op}}$ and $A_{\mathrm{op}}$ for $\mathcal{T}_{\mathrm{op}}$

In *Phase 1*, Subroutine 3 identifies the ancestors in $G^{\mathrm{J}}$ using the pairwise distances $d_{ij}$ for all $i, j \in V^{\mathrm{O}}$. In this phase, it returns a collection $\mathfrak{V}$ of vertex triplets such that each triplet collection $\mathcal{V} \in \mathfrak{V}$ contains (and only contains) all vertex triples that share an identical ancestor in $G^{\mathrm{J}}$. The key component for this is to use TIA (Test Identical Ancestor). In the *Phase 2*, Subroutine 3 enrolls each collection in $\mathfrak{V}$ to either $\mathfrak{V}_{\mathrm{obs}}$ or $\mathfrak{V}_{\mathrm{hid}}$, such that $\mathfrak{V}_{\mathrm{obs}}$ ($\mathfrak{V}_{\mathrm{hid}}$) contains the collection of vertex triplets for which their corresponding ancestors are observed (hidden resp.), and observed ancestors are enrolled in the set $A_{\mathrm{obs}}$. For identifying the observed ancestors from $\mathfrak{V}$, Subroutine 3 does the following for each collection $\mathcal{V} \in \mathfrak{V}$: it checks for a vertex triplet $T$ in $\mathcal{V}$ for which one vertex in the triplet $T$ separates the other two. In the final phase, Subroutine 3 accepts $d_{ij}$ for each pair $i, j \in V^{\mathrm{O}}$ and $\mathfrak{V}_{\mathrm{hid}}$, and learns the pairwise distance $d_{ij}$ for each $i \in V^{\mathrm{O}}$ and $j \in A_{\mathrm{hid}}$ by finding a vertex triplet $T$ in a collection $\mathcal{V}_j \in \mathfrak{V}_{\mathrm{hid}}$ such that $T$ contains $i$. Then, in the next step, Subroutine 3 learns $d_{ij}$ for each $i, j \in A_{\mathrm{hid}}$ by selecting the most frequent distance in $\Delta_{pq}$ as defined in Section 3.4.

We next present Subroutine 4 for clustering the vertices in the set $V^{\mathrm{O}} \setminus A_{\mathrm{obs}}$. It accepts $A_{\mathrm{obs}}$, $A_{\mathrm{hid}}$, and $\{d_{ij}\}_{i \in V^{\mathrm{O}}, j \in A_{\mathrm{hid}}}$, and enrolls each vertex in $V^{\mathrm{O}} \setminus A_{\mathrm{obs}}$ either in a *leaf cluster* (see Phase 1 of Subroutine 4) or in an *internal cluster* ( see Phase 2 of Subroutine 4). Now, for the collection of leaf clusters $\mathcal{L}$, each cluster $L \in \mathcal{L}$ is associated to a unique element $a \in A$ such that $L_2$ is separated from $A \setminus a$ by $a$. Each cluster $I \in \mathcal{I}$ is associated with a subset of ancestors $I_1 \subset A$, such that $I_2$ is separated from all other ancestors in $A \setminus I_1$ by $I_1$.

**Procedure 2** TestIdenticalAncestor (TIA)

**procedure** TIA($U, W$)
  **if** for all $x \in U$, $\exists$ at least a pair $y, z \in W$ such that $d_x^U + d_y^W = d_{xy}$ and $d_x^U + d_z^W = d_{xz}$ **then**
    **Return** True.
  **end if**
  **Return** False.
**end procedure**

**Procedure 7** NonBlockNeighbors

**Input:** An ancestor vertex $u$, $\mathcal{C}_u$, $A_{\mathrm{op}}$, and the extended distance set $\mathcal{D}_{\mathrm{ext}}$.
**Output:** Neighbors $\delta(u)$ of $u$ such that they do not belong to the clusters that contains $u$.
**Initialize:** $\delta(u) \triangleq A_{\mathrm{op}} \setminus \bigcup_{C \in \mathcal{C}_u} C_3$.
**for** each $x \in \delta(u)$ **do**
  **if** $\exists$ a vertex $b \in \mathcal{C}_u \setminus$ s.t. $d_{ux} = d_{ub} + d_{bx}$ **then**
    $\delta(u) \leftarrow \delta(u) \setminus x$
  **end if**
**end for**
**for** each $k, \ell \in \binom{\delta(u)}{2}$ **do**
  **if** $d_{uk} + d_{k\ell} = d_{u\ell}$ **then**
    $\delta(u) \leftarrow \delta(u) \setminus \ell$.
  **end if**
**end for**

**Subroutine 8** Learning $E_{\mathrm{op}}$ for $\mathcal{T}_{\mathrm{op}}$

**Input:** The collection of leaf clusters $\mathcal{L}$ and internal clusters $\mathcal{I}$, $\mathcal{C} \triangleq \mathcal{L} \cup \mathcal{I}$, a subset $\mathcal{E}_{\mathrm{leaf}}$ of $E_{\mathrm{op}}$.
**Output:** An edge set $E_{\mathrm{op}}$ for $\mathcal{T}_{\mathrm{op}}$.
**Initialize:** $E_{\mathrm{op}} \leftarrow \mathcal{E}_{\mathrm{leaf}}$.
**for** each $u \in A_{\mathrm{op}}$ **do**
  Let $C \in \mathcal{C}$ be the cluster such that $C_3 \ni u$
  Get $\delta(u)$ from NonBlockNeighbors($u, C, A_{\mathrm{op}}$).
  **for** each $P_u \in \mathcal{P}_{\mathrm{op}}$ s.t. $P_u \ni u$ **do**
    **for** each $v \in \delta(u)$ **do**
      $E_{\mathrm{op}} \leftarrow E_{\mathrm{op}} \cup (P_u, \{v\}, u, v)$
    **end for**
  **end for**
**end for**
**Return** The edge set $E_{\mathrm{op}}$ for $\mathcal{T}_{\mathrm{op}}$.

We now discuss NonCutTest appears in Procedure 5. The goal of NonCutTest is to learn (a) the non-cut vertices, and (b) potential cut vertices of a non-trivial block from a leaf cluster. NonCutTest accepts a set $W \subseteq V^{\mathrm{O}}$ s.t. $|W| \geq 3$, and partitions the vertex set $W$ into $C_{\mathrm{cut}}$ (the set of potential cut vertices) and $C_{\mathrm{non-cut}}$ (the set of vertices which *can not* be a cut vertex). Then we use Subroutine 6 for learning (a) vertex

**Subroutine 3** Identifying Ancestors and Extending the Pairwise Distance Set

**Input:** Pairwise distances $\mathcal{D} = \{d_{ij}\}_{i,j \in V^{\mathrm{O}}}$, where $V^{\mathrm{O}}$ is the set of observed vertices.
**Return:** A collection of vertex triplets $\mathfrak{V}_{\mathrm{obs}}$ ($\mathfrak{V}_{\mathrm{hid}}$ resp.) with observed (hidden resp.) ancestors, the set $A_{\mathrm{obs}}$ ($A_{\mathrm{hid}}$ resp.) of observed (hidden resp.) ancestors, the set of pairwise distances $\{d_{ij}\}$ for each pair $i, j \in V^{\mathrm{O}} \cup A_{\mathrm{hid}}$.
**Initialize:** $\mathfrak{V}_{\mathrm{obs}}, \mathfrak{V}_{\mathrm{hid}}, \widetilde{\mathcal{D}}, \mathcal{D}_{\mathrm{hid}} \leftarrow \emptyset$, collection of vertex triplets $\mathcal{V} \triangleq \binom{V^{\mathrm{O}}}{3}$, counter $n = 1$

*Phase 1 – Clustering Star Triplets*

**for** each $U \in \mathcal{V}$ **do**
  $\mathcal{V}_n \triangleq \{W \subset \mathcal{V} : \text{TIA}\ (U, W) \text{ is TRUE}\} \cup U$
  **if** $|\mathcal{V}_n| > 1$ **then** $n = n + 1$
  **end if**
  $\mathfrak{V} \leftarrow \mathfrak{V} \cup \mathcal{V}_n$ ▷ enrolling the collection $\mathcal{V}_n$ to $\mathfrak{V}$
**end for**
**Return:** $\mathfrak{V} = \{\mathcal{V} \subset \binom{V}{3} : \text{each } U \in \mathcal{V} \text{ has the same ancestor }\}$,

*Phase 2 – Labeling Ancestors*

**for** each collection $\mathcal{V} \in \mathfrak{V}$ **do**
  **if** $\exists$ a triplet $V \triangleq \{u, v, w\} \in \mathcal{V}$ s.t. $d_{uv} + d_{vw} = d_{uw}$ **then**
    $\mathfrak{V}_{\mathrm{obs}} \leftarrow \mathfrak{V}_{\mathrm{obs}} \cup \mathcal{V}$
    $A_{\mathrm{obs}} \leftarrow A_{\mathrm{obs}} \cup v$
  **else**
    $\mathfrak{V}_{\mathrm{hid}} \leftarrow \mathfrak{V}_{\mathrm{hid}} \cup \mathcal{V}$
  **end if**
**end for**
Set $A_{\mathrm{hid}} \triangleq \{a_i | i \in [|\mathfrak{V}_{\mathrm{hid}}|]\}$ ▷ introduce one vertex for each element in $\mathfrak{V}_{\mathrm{hid}}$

*Phase 3 – Learning the pairwise distance set $\{d_{ij}\}_{i,j \in V^{\mathrm{O}} \cup A_{hid}}$*

**for** each $\mathcal{V}_i \in \mathfrak{V}_{\mathrm{hid}}$ **do**
  **for** each $j \in V^{\mathrm{O}}$ **do**
    Find a triplet $U \in \mathcal{V}_i$ s.t $U \ni j$ ▷ cf. Claim 3
    $\widetilde{\mathcal{D}} \leftarrow \widetilde{\mathcal{D}} \cup \left((i, j), d_j^U\right)$
  **end for**
**end for**
**for** each $p \neq q \in A_{\mathrm{hid}}$ **do**
  Pick a pair of triplets $U_p \in \mathcal{V}_p, U_q \in \mathcal{V}_q$.
  $\Delta_{pq} = \left\{d_{xy} - (d_x^{U_p} + d_y^{U_r}) : x \in U_p, y \in U_q\right\}$.
  $\mathcal{D}_{\mathrm{hid}} \leftarrow \mathcal{D}_{\mathrm{hid}} \cup \left((p, q), \mathrm{mode}(\Delta_{pq})\right)$ ▷ most frequent element in $D_{pq}$
**end for**
**Return** $\mathfrak{V}_{\mathrm{obs}}$, $\mathfrak{V}_{\mathrm{hid}}$, $A_{\mathrm{obs}}$, $A_{\mathrm{hid}}$, $\widetilde{\mathcal{D}}$, and $\mathcal{D}_{\mathrm{hid}}$.

**Subroutine 4** LEARNCLUSTERS

**Input:** $A_{\mathrm{obs}}, A_{\mathrm{hid}}$, and $\mathcal{D}$, and $A \triangleq A_{\mathrm{obs}} \cup A_{\mathrm{hid}}$.
**Output:** A collection of leaf clusters $\mathcal{L}$ and internal clusters $\mathcal{I}$.
**Initialize**: $\mathcal{L} \triangleq (L_1, L_2, L_3), \mathcal{I} \triangleq (I_1, I_2, I_3)$, and $L_1, L_2, L_3, I_1, I_2, I_3 \leftarrow \emptyset$.

*Phase 1 – Learning Leaf Clusters*

**for** each $x \in V^{\mathrm{O}} \setminus A_{\mathrm{obs}}$ **do**
  **if** $\exists a \in A$ such that $d_{xa} + d_{aa'} = d_{xa'}$ for all $a' \in A \setminus \{a\}$ **then**
    **if** $\exists\ L \in \mathcal{L}$ such that $L_1 = a$ **then**
      $L_2 \leftarrow L_2 \cup \{x\}$
    **else**
      $L \triangleq (a, \{x\}, \emptyset)$
      $\mathcal{L} \leftarrow \mathcal{L} \cup L$
    **end if**
    $V^{\mathrm{O}} \leftarrow V^{\mathrm{O}} \setminus \{x\}$
  **end if**
**end for**
**Return** $\mathcal{L} = \{L : L_2 \in 2^{V^{\mathrm{O}} \setminus A_{\mathrm{obs}}}$
s.t. $L_2$ is separated from $A \setminus L_1$ by $L_1$ where $|L_1| = 1\}$.

*Phase 2 – Learning Internal Clusters*

**for** each $x \in V^{\mathrm{O}} \setminus A_{\mathrm{obs}}$ **do**
  **for** each $\tilde{A} \subset 2^A$ s.t. $|\tilde{A}| > 1$ **do**
    **for** each pair $k, \ell \in \binom{\tilde{A}}{2}$ **do**
      **if** there exists a pair $(k, \ell)$ s.t. $d_{xk} + d_{k\ell} = d_{x\ell}$ or $d_{x\ell} + d_{\ell k} = d_{xk}$ **then**
        **end if**
        **Break**
    **end for**
  **end for**
  **if** $\exists$ a $I \in \mathcal{I}$ such that $I_1 = \tilde{A}$ **then**
    $I_2 \leftarrow I_2 \cup \{x\}$.
  **else**
    $I \triangleq (\tilde{A}, \{x\}, \emptyset)$
    $\mathcal{I} \leftarrow \mathcal{I} \cup I$
  **end if**
**end for**
**Return** $\mathcal{I} = \{I : I_2 \in 2^{V^{\mathrm{O}} \setminus A_{\mathrm{obs}}}$
s.t. $I_2$ is separated from $A \setminus I_1$ by $I_1$ where $|I_1| > 1\}$.

**Procedure 5** NONCUTTEST

**Input:** A leaf cluster $L \in \mathcal{L}$ such that $|L_2| \geq 2$.
**Output:** A set $C_{\mathrm{cut}}, C_{\mathrm{non-cut}} \triangleq L_2 \setminus C_{\mathrm{cut}}, L_3 \subseteq L_2$.
**Initialize:** $C_{\mathrm{cut}}$ with $L_2$.
**for** each $x \in L_2$ **do**
  **for** each pair $y, z \in L_2 \setminus \{x\}$ **do**
    Pick any arbitrary pair $\alpha_1, \alpha_2 \in V^{\mathrm{O}} \setminus L_2$.
    $U_i \triangleq \{x, y, \alpha_1\}$ and $U_j \triangleq \{x, z, \alpha_2\}$.
    **if** $\text{TIA}(U_i, U_j)$ is FALSE **then**
      **end if**
      **Break**
    $C_{\mathrm{cut}} \leftarrow C_{\mathrm{cut}} \setminus \{x\}$ ▷ $x$ is not a non-cut vertex.
  **end for**
**end for**
**if** $(|C_{\mathrm{cut}}|) > 1 \wedge \left(L_1 \notin A_{\mathrm{obs}}\right)$ **then**
  Pick an arbitrary vertex $a$ from $C_{\mathrm{cut}}$ and set $L_3 \triangleq a$.
**end if**
**Return** $C_{\mathrm{cut}}$, $C_{\mathrm{non-cut}}$, and $L_3$.

**Subroutine 6** Partitioning and learning local edges (PALE)

**Input:** The observed vertex set $V^{\mathrm{O}}$, the collection of leaf clusters $\mathcal{L}$ and internal clusters $\mathcal{I}$.
**Output:** The vertex set $\mathcal{P}_{\mathrm{op}}$, the articulation points $A_{\mathrm{op}}$, and a subset $\mathcal{E}_{\mathrm{leaf}}$ of the edge set $E_{\mathrm{op}}$ for $\mathcal{T}_{\mathrm{op}}$. Each element $E \in \mathcal{E}_{\mathrm{leaf}}$ is an ordered quadruple such that $E_1, E_2 \subseteq V^{\mathrm{O}}$, and $E_3 \in E_1, E_4 \in E_2$.
**Initialize** $\mathcal{P}_{\mathrm{op}}, A_{\mathrm{op}}, \mathcal{E}_{\mathrm{leaf}} \leftarrow \emptyset$.
$A_{\mathrm{cluster}} \triangleq \{c \in L_1 : c \in A_{\mathrm{obs}}\}$
$\mathcal{P}_{\mathrm{op}} \leftarrow \mathcal{P}_{\mathrm{op}} \cup (A_{\mathrm{obs}} \setminus A_{\mathrm{cluster}})$, $A_{\mathrm{op}} \triangleq A_{\mathrm{obs}}$.

*Phase 1 – Partitioning and Local Edge Learning w.r.t. the Leaf Clusters*

**for** each $L \in \mathcal{L}$ s.t. $(|L_2| < 3) \wedge (L_1 \notin A_{\mathrm{obs}})$ **do** ▷ ancestor in the leaf cluster is not observed.
  Pick an arbitrary vertex $a \in L_2$, $L_3 \leftarrow a$, $L_2 \leftarrow L_2 \setminus \{a\}$, $A_{\mathrm{op}} \leftarrow A_{\mathrm{op}} \cup \{a\}$.
  $\mathcal{P}_{\mathrm{op}} \leftarrow \mathcal{P}_{\mathrm{op}} \cup L_2 \cup L_3$
  $\mathcal{E}_{\mathrm{leaf}} \leftarrow \mathcal{E}_{\mathrm{leaf}} \cup (L_2, L_3, L_2, L_3)$
**end for**
**for** each $L \in \mathcal{L}$ s.t. $(|L_2| \geq 3) \wedge (L_1 \notin A_{\mathrm{obs}})$ **do** ▷ ancestor in the leaf cluster is not observed.
  Get $C_{\mathrm{cut}}, C_{\mathrm{non-cut}}$ and $L_3$ from NonCutTest($L$).
  $A_{\mathrm{op}} \leftarrow A_{\mathrm{op}} \cup L_3$
  Set $B \triangleq C_{\mathrm{non-cut}} \cup L_3$ ▷ $C_{\mathrm{non-cut}}$ and $L_3$ contains the non-cut vertices and cut vertex, respectively.
  $\mathcal{P}_{\mathrm{op}} \leftarrow \mathcal{P}_{\mathrm{op}} \cup B \cup \bigcup_{v \in C_{\mathrm{cut}} \setminus L_3} \{v\}$ ▷ $C_{\mathrm{cut}}$ can contain multiple cut vertices.
  $\mathcal{E}_{\mathrm{leaf}} \leftarrow \mathcal{E}_{\mathrm{leaf}} \cup \bigcup_{v \in C_{\mathrm{cut}} \setminus L_3} (B, \{v\}, L_3, v)$
**end for**
**for** each $L \in \mathcal{L}$ s.t. $(|L_2| = 1) \wedge (L_1 \in A_{\mathrm{obs}})$ **do** ▷ ancestor in the leaf cluster is observed.
  $\mathcal{P}_{\mathrm{op}} \leftarrow \mathcal{P}_{\mathrm{op}} \cup L_2 \cup L_3$
  $\mathcal{E}_{\mathrm{leaf}} \leftarrow \mathcal{E}_{\mathrm{leaf}} \cup (L_2, L_3, L_2, L_3)$
**end for**
**for** each $L \in \mathcal{L}$ s.t. $(|L_2| > 1) \wedge (L_1 \in A_{\mathrm{obs}})$ **do** ▷ ancestor in the leaf cluster is observed.
  Get $C_{\mathrm{cut}}, C_{\mathrm{non-cut}}$ and $L_3$ from NonCutTest($L$)
  Set $B \triangleq C_{\mathrm{non-cut}} \cup L_1$.
  $\mathcal{P}_{\mathrm{op}} \leftarrow \mathcal{P}_{\mathrm{op}} \cup B \cup \bigcup_{v \in C_{\mathrm{cut}}} \{v\}$.
  $\mathcal{E}_{\mathrm{leaf}} \leftarrow \mathcal{E}_{\mathrm{leaf}} \cup \bigcup_{v \in C_{\mathrm{cut}}} (B, \{v\}, L_3, v)$
**end for**

*Phase 2 – Partitioning w.r.t. the Internal Clusters*

**for** each $I \in \mathcal{I}$ **do**
  **for** each $i \in I_1$ **do**
    **if** $i \notin A_{\mathrm{obs}}$ **then**
      Find the $L \in \mathcal{L}$ s.t. $L_1 = i$
      $I_3 \leftarrow I_3 \cup i$, $A_{\mathrm{op}} \leftarrow A_{\mathrm{op}} \cup \{i\}$
    **end if**
  **end for**
  $\mathcal{P}_{\mathrm{op}} \leftarrow \mathcal{P}_{\mathrm{op}} \setminus I_3$
  $B \triangleq \{I_2 \cup I_3\}$ and $\mathcal{P}_{\mathrm{op}} \leftarrow \mathcal{P}_{\mathrm{op}} \cup B$. ▷ $B$ is an internal non trivial block.
**end for**
**Return** $\mathcal{P}_{\mathrm{op}}$, $A_{\mathrm{op}}$, and a subset $\mathcal{E}_{\mathrm{leaf}}$ of $E_{\mathrm{op}}$.

set $\mathcal{P}_{\mathrm{op}}$, (b) articulation points $A_{\mathrm{op}}$, and a subset of the edge set $E_{\mathrm{op}}$ for $\mathcal{T}_{\mathrm{op}}$. The Subroutine 6 learns (a), (b), and (c) from both leaf clusters and internal clusters. In the following, we list all the possible cases of leaf clusters Subroutine 6 considered in learning $\mathcal{P}_{\mathrm{op}}$ and $A_{\mathrm{op}}$: Leaf clusters contains 1. At most two vertices with hidden ancestor, 2. More than two vertices with hidden ancestor, 3. One vertex with observed ancestor, and 4. More than one vertex with observed ancestor. For each $I \in \mathcal{I}$, Subroutine 6 checks whether $i \in A_{\mathrm{obs}}$. If $i \notin A_{\mathrm{obs}}$, then the subroutine finds the leaf cluster $L$ s.t. $L_1 \ni i$.

### A.2 Learning $E_{\mathrm{op}}$ for $\mathcal{T}_{\mathrm{op}}$

The next goal of NoMAD is to learn to learn the edge set $E_{\mathrm{op}}$ for $A_{\mathrm{op}}$. Precisely, NoMAD learns the neighbors of each articulation point in $A_{\mathrm{op}}$. The learning of $E_{\mathrm{op}}$ is divided into two steps: (a) Learn the neighbors of each articulation points (appears in Procedure 3), and (b) use the information obtained from (a) to construct $E_{\mathrm{op}}$ (appears in Procedure 4).

## B Theory: Guaranteeing the Correctness of the NoMAD

In this section we will prove that NoMAD correctly learns the equivalence class. We star this section by restating Theorem 4.2 from Section 4.

**Theorem B.1.** *Consider a covariance matrix $\Sigma^*$ whose conditional independence structure is given by the graph $G$, and the model satisfies Assumption 4.1. Suppose that according to the problem setup in Section 3.1, we are given pairwise distance $d_{ij}$ of a vertex pair $(i, j)$ in the observed vertex set $V^O$, that is, $d_{ij} \triangleq -\log|\rho_{ij}|$ where $\rho_{ij} \triangleq \Sigma^o_{ij}/\sqrt{\Sigma^o_{ii}\Sigma^o_{jj}}$. Then, given the pairwise distance set $\{d_{ij}\}_{i,j \in V^o}$ as inputs, NoMAD outputs the equivalence class $[G]$.*

**Proof Outline.** We show that NoMAD correctly learns the equivalence class by showing that it can correctly learn $\mathcal{T}_{\mathrm{op}}$. Given this, and using Definition 3.2, the entire equivalence class can be readily generated. We show that NoMAD learns $\mathcal{T}_{\mathrm{op}}$ correctly by proving that (a) the vertex set $\mathcal{P}_{\mathrm{op}}$, (b) the articulation points $A_{\mathrm{op}}$, and (c) the edge set $E_{\mathrm{op}}$ are learnt correctly. Following is the outline for (a) and (b). From Section 3.3, it is clear that NoMAD succeeds in finding the ancestors, which is the first step, provided the TIA tests succeed (established in Lemma B.7).Then, Proposition B.13 establishes that NoMAD learns $\mathcal{P}_{\mathrm{op}}$ and $A_{\mathrm{op}}$ correctly. The proof correctness of this step crucially depends on identifying the non-cut vertices (c.f. Lemma B.12).

Then, for establishing the correctness of NoMAD in learning $E_{\text{op}}$, NoMAD learns the neighbor articulation points of each articulation point. Proposition B.14 shows that NoMAD correctly learns $E_{\text{op}}$.

**Lemma B.1.** *Let $G$ be a graph on vertex set $V$, and $\mathcal{T}_{\text{AST}}(G)$ be the corresponding articulated set tree of $G$. Then, $\mathcal{T}_{\text{AST}}(G)$ is a tree.*

*Proof.* We prove $\mathcal{T}_{\text{AST}}(G)$ is a tree by showing that $\mathcal{T}_{\text{op}}$ is connected and acyclic. Suppose on the contrary that $\mathcal{T}_{\text{AST}}(G)$ contains a cycle $B'$. Then, $B'$ is a non-trivial block in $G$ with no cut vertex. This would contradict the maximality of the non-trivial blocks contained in the cycle $B'$. Hence, any cycle is contained in a unique non-trivial block in $\mathcal{T}_{\text{AST}}(G)$. We now show that $\mathcal{T}_{\text{op}}$ is connected. Recall that vertices in $\mathcal{T}_{\text{AST}}(G)$ can either be a non-trivial block or a singleton vertices not part of any non-trivial block. Consider any vertex pair $(u, v)$ in $\mathcal{T}_{\text{AST}}(G)$. We will find a path from $u$ to $v$. Suppose that $u$ and $v$ are non-singletons, and associated with non-trivial blocks $B_u$ and $B_v$ respectively. Since, $G$ is connected, $\exists$ a path between the articulation points of $B_u$ and $B_v$. Hence, $u$ and $v$ are connected in $\mathcal{T}_{\text{AST}}(G)$. The other cases where one of them is a singleton vertex or both are singleton vertices follows similarly.

□

We now show that NoMAD correctly learns $[G]$. For the graph $G$ on a vertex set $V$, let $G^{\text{J}} = (V^{\text{J}}, E^{\text{J}})$ be defined as in Subsection 3.1. Let $A^{\text{J}}$ be the set of ancestors in $G^{\text{J}}$. Recall that NoMAD only observes samples from a subset $V^{\text{O}} \subseteq V^{\text{J}}$ of vertices. NoMAD uses $\{d_{ij}\}_{i,j \in V^{\text{O}}}$ to learn $\mathcal{T}_{\text{op}}$, which in turn will output $[G]$. Hence, each theoretical section first states a result of $G^{\text{J}}$ *assuming* that the pair $(V^{\text{J}}, E^{\text{J}})$ is known.

**Correctness in Learning $\mathcal{P}_{\text{op}}$ and $A_{\text{op}}$.** We first establish that NoMAD correctly identifies ancestors in $G^{\text{J}}$. In the following, we first identify the vertices in $G$ which are ancestors in $G^{\text{J}}$. Then, in Lemma B.5, we show the existence of at least two vertex triplets for each ancestor in $G^{\text{J}}$. Finally, in Proposition B.10, we show that Subroutine 3 correctly identifies the star triplets in $G^{\text{J}}$. We start with introducing $uw$-separator.

**Definition B.2** ($uw$-separator)**.** *Consider an arbitrary pair $u, w \in V$ in the graph $G$. We say $v \in V \setminus \{u, w\}$ is a $uw$- separator in $G$ if and only if any path $\pi \in \mathcal{P}_{uw}$ contains $v$.*

**Lemma B.3.** *A vertex $a \in V^{\text{J}}$ is an ancestor in $G^{\text{J}}$ if and only if $a$ is an $uw$- separator in $G$, for some $u, w \in V$.*

*Proof.* ($\Rightarrow$) Suppose $a \in V^{\text{J}}$ is an ancestor in $G^{\text{J}}$. Then, we show that $a$ is an $uw$- separator in $G$. Fix a triplet $T \triangleq \{a_1^e, a_2^e, a_3^e\} \in \mathcal{V}_a$, where $\mathcal{V}_a \triangleq$ collection of all triplets with ancestor $a$ in $G^{\text{J}}$. Then, any path $\pi \in \mathcal{P}_{a_i^e a_j^e}$ contains $a$ in $G^{\text{J}}$, for $i, j = 1, 2,$ and 3. Thus, $a_i^e \perp\!\!\!\perp a_j^e | a$, and $a$ is an $uw$-separator with $u = a_i^e,$ and $w = a_j^e$. Furthermore, for a joint graph following is true for any vertex $u$ and its corresponding noisy samples $u^e$: $u^e \perp\!\!\!\perp v | u$ for all $v \in V^{\text{J}} \setminus \{u, u^e\}$. Hence, we can conclude that $a_i \perp\!\!\!\perp a_j \mid a$, and $a$ is an $uw$-separator in $G$.

($\Leftarrow$) Suppose that $\exists\, u, w \in V$ for which $a \in V$ is an $uw$- separator in $G$. Then, we show that $v$ is an ancestor in $G^{\text{J}}$ by constructing a triplet $T$ with ancestor $a$. This construction directly follows from Definition B.2 and Definition 3.4.

□

**Lemma B.4.** *Let $V_{\text{cut}}$ be the set of all cut vertices in $G$. Then, there does not exist any pair $u, w \in V$ such that $b \in V \setminus V_{\text{cut}}$ is an $uw$- separator in $G$.*

*Proof.* Let $b \in V \setminus V_{cut}$. Then, notice that $b$ can be either (1) a non-cut vertex of a non-trivial block or a (2) leaf vertex in $G$. For (1), by the definition of a block, any non-cut vertex ceases to be a $uw$-separator for any $u \neq b$ and $w \neq b$ in $V$. For (2), since $b$ is a leaf vertex, its degree is one, and hence, cannot be a $uw$-separator for any $u \neq b$ and $w \neq b$ in $V$. □

**Lemma B.5.** *Let $A^{\text{J}}$ be the set of all ancestors in $G^{\text{J}}$. Then, for each $a \in A^{\text{J}}$, there exists at least two triplets $U, W \in \binom{V^{\text{O}}}{3}$ for which $a$ is the ancestor in $G^{\text{J}}$.*

*Proof.* We construct two triplets for any ancestor in $G^{\text{J}}$. Lemma B.4 states that only a cut-vertex in $G$ is an ancestor in $G^{\text{J}}$. First, let $c$ be a cut-vertex of a non-trivial block $B$ in $G$. Pick any two non-cut vertices $x, y \in B \setminus \{c\}$. Then, consider the following two triplets in $V^{\text{O}}$ : $\{x^e, c^e, \alpha_1^e\}$ and $\{y^e, c^e, \alpha_2^e\}$, where $\alpha_1, \alpha_2 \in V \setminus B$. Then both $\{x^e, c^e, \alpha_1^e\}$ and $\{y^e, c^e, \alpha_2^e\}$ share the ancestor $c$ in $G^{\text{J}}$. Now, let $c$ be a cut vertex which is not in any non-trivial block. Consider two blocks $B_i$ and $B_j$ such that $B_i \perp\!\!\!\perp B_j \mid c$. Then, consider the following pair: $\{i_1, c, j_1\}$ and $\{i_2, c, j_2\}$ s.t. $i_1, i_2 \in B_i$ and $j_1, j_2 \in B_j$. Notice that $\{i_1^e, c^e, j_1^e\}$ and $\{i_2^e, c^e, j_2^e\}$ in $\binom{V^{\text{O}}}{3}$ share the ancestor $c$ in $G^{\text{J}}$. Finally, if $G$ is a tree on three vertices, then $G$ has an unique ancestor.

□

**Claim 1.** *Let (a) $\{i, j, k\}$ be a vertex triple in $G$, and (b) $i^e$ be the corresponding noisy counterpart of $i$. Then, $j$ separates $i$ and $k$ if and only if $j$ separates $i^e$ and $k$ in combined graph $G^{\text{J}}$*

*Proof.* The forward implication directly follows from the construction of joint graph. For the reverse implication suppose that in $G^{\text{J}}$, $i^e \perp\!\!\!\perp k|j$. We show that this implies $i \perp\!\!\!\perp k|j$ and $k$ in $G$. Suppose on the contrary that $i \not\perp\!\!\!\perp k|j$. That means $\exists$ a path $\pi$ between $i$ and $k$ that does not contain $j$. Now, notice that $\pi \cup \{i, i^e\}$ is a valid path between $i^e$ and $k$ in $G^{\text{J}}$ that does not contain $j$, and it violates the hypothesis.

□

The following lemma relates an observed ancestor in a triplet $T$ with the remaining pair.

**Lemma B.6.** *Suppose that a triplet $T \in \binom{V^{\text{O}}}{3}$ is a star triplet in $G^{\text{J}}$. A vertex $v \in T$ is an $uw-$ separator for $u, w \in T \setminus v$ if and only if $v$ is the ancestor of $T$.*

*Proof.* Suppose that a vertex $v \in T$ is an $uw-$ separator for $u, w \in T \setminus v$. We show that $v$ is an ancestor. As $v$ is an $uw-$ separator, i.e., $u \perp\!\!\!\perp w|v$. Suppose on the contrary that $v' \neq v$ is the ancestor of $T$ in $G^{\text{J}}$. We show that $v$ is not an $uw-$ separator for $u, w \in T \setminus v$. As $v'$ is the ancestor of $T$, $u \perp\!\!\!\perp w|v'$. (according to Definition 3.4). This contradicts the hypothesis that $u \perp\!\!\!\perp w|v$. Thus, $v$ and $v'$ are identical. Therefore, $v$ is the ancestor of $\{u, v, w\}$. The reverse implication follows from Definition 3.4. □

We will now prove the correctness of the TIA test. We proceed with the following claim.

**Claim 2.** *Suppose that $U$ and $W \in \binom{V^{\text{O}}}{3}$ are star triplets with non-identical ancestors $r_u$ and $r_w$, resp. Then, there exists a vertex $u \in U$ and a pair, say $w_2, w_3 \in W$, such that all paths $\pi \in \mathcal{P}_{uw_i}$ for $i = 1, 2$ contain both $r_u$ and $r_w$.*

*Proof.* Without loss of generality, let $W = \{w_1, w_2, w_3\}$. We prove this claim in two stages. In the first stage, we show that for each vertex $u \in U$ there exists at least a pair $w_2, w_3 \in W$ such that $u \perp\!\!\!\perp \{w_2, w_3\}|r_w$. Then, in the next stage we show that there exists a vertex $u \in U$ such that $u \perp\!\!\!\perp r_w|r_u$. For the first part, suppose on the contrary that there exists a vertex $u \in U$ and a pair $w_2, w_3 \in W$ such that there exists a path $\pi_2 \in \mathcal{P}_{uw_2}$ and a path $\pi_3 \in \mathcal{P}_{uw_3}$ such that $r_w \notin \pi_2$ and $r_w \notin \pi_3$. Then, one can construct a path between $w_2$ and $w_3$ that does not contain $r_w$, which violates the hypothesis that $W$ is a star triplet. Now, in the next step of proving the claim, we show that there exists a vertex $u \in U$ such that $u \perp\!\!\!\perp r_w|r_u$. Now, suppose that for all $u \in U$ there exists a path between $u$ and $r_w$, that does not contain $r_u$. We will next show that this implies there has to be a path between $u_1$ and $u_2$ ($u_1, u_2 \in U$) that does not include $r_u$. We will show this constructively. Let $s$ be the last vertex in the path $\pi_{u_1 r_w}$ that is also contained in $\pi_{u_2 r_w}$. Note that $\pi_{u_1 s}$ and $\pi_{u_2 s}$ are valid paths in the graph, and that their concatenation is a valid path between $u_1$ and $u_2$. This proves that $u_1$ and $u_2$ are connected by a path that is not separated by $r_u$, and hence contradicting the hypothesis that $U$ is a star triplet. Finally, let $u' \in U$ be the vertex for which $u' \perp\!\!\!\perp r_w \mid r_u$. Then, there exists a triplet $\{u', w_2, w_3\}$ such that both $r_u$ and $r_w$ separates $u'$ and $w_2$, and both $r_u$ and $r_w$ separates $u'$ and $w_3$. □

Using Claim 2, we now show the correctness of our TIA test. Recall that the TIA $(U, W)$ accepts triplets $U, W \in \binom{V^{\text{O}}}{3}$, and returns TRUE if and only if $U$ and $W$ share an ancestor in $G^{\text{J}}$. Also recall the following assumption: Let $U, W \in \binom{V}{3} \setminus \mathcal{V}_{\text{star}} \cup \mathcal{V}_{\text{sep}}$. Then, (i) there are no vertices $x \in U$ and $a \in W$ that satisfy

$d_x^U + d_a^W = d_{xa}$, and (ii) there does not exist any vertex $r \in V$ and $x \in U$ for which the distance $d_{xr}$ satisfies relation in equation 1.

**Lemma B.7.** ***(Correctness of TIA test)*** *Fix any two vertex triplets $U \neq W \in \binom{V^o}{3}$. TIA$(U, W)$ returns* TRUE *if and only if $U$ and $W$ are star triplets in $G^{\text{J}}$ with an identical ancestor $r \in V$.*

*Proof.* From Subroutine 2 returning TRUE is same as checking that for all $x \in U$, there exist at least two vertices $y, z \in W$ such that both of the following hold

$$d_x^U + d_y^W = d_{xy}, \tag{2}$$

$$d_x^U + d_z^W = d_{xz}. \tag{3}$$

Suppose that $U$ and $W$ are star triplets with an identical ancestor $r \in V$. We prove by contradiction. Let $a \in U$ and assume that there is *at most* one vertex $x \in V$ such that $d_a^U + d_x^W = d_{ax}$. Therefore, one can find two vertices $y_1, y_2 \in V$ such that

$$d_a^U + d_{y_i}^W \neq d_{ay_i}, \quad i = 1, 2. \tag{4}$$

However, from our hypothesis that $U$ and $W$ are star triplets with the common ancestor $r$, we know that $d_a^U = d_{ar}$ and $d_{y_i}^W = d_{ry_i}$, for $i = 1, 2$. This, along with equation 4, implies that $r$ does not separate $a$ from $y_1$ or $y_2$. For $i = 1, 2$, let $\pi_{ay_i}$ be the path between $a$ and $y_i$ that does not include $r$. We will next show that this implies there has to be a path between $y_1$ and $y_2$ that does not include $r$. We will show this constructively. Let $s$ be the last vertex in the path $\pi_{ay_1}$ that is also contained in $\pi_{ay_2}$. Note that $\pi_{y_1 s}$ and $\pi_{sy_2}$ are valid paths in the graph, and that their concatenation is a valid path between $y_1$ and $y_2$. This proves that $y_1$ and $y_2$ are connected by a path that is not separated by $r$, and hence contradicting the first hypothesis.

For the reverse implication, we do a proof by contrapositive. Fix two triplets $U$ and $W$. Suppose that $U$ and $W$ are *not star triplets with an identical ancestor* in $G^{\text{J}}$. We will show that this implies that there exists at least one vertex in $U$ for which no pair in $W$ satisfies both Eq. equation 2 and Eq. equation 3. To this end, we will consider all three possible configurations for a triplet pair $U$ and $W$ where they are not star triplets with an identical ancestor in $G^{\text{J}}$: 1. $U$ and $W$ are star triplets with a non-identical ancestor in $G^{\text{J}}$, 2. Both $U$ and $W$ are non-star triplets in $G^{\text{J}}$, 3. $U$ is a star triplet and $W$ is a non-star triplet in $G^{\text{J}}$. Then, for each configuration, we will show that there exists at least a vertex $x \in U$ for which no pair in $W$ satisfies both Eq. equation 2 and Eq. equation 3

**$U$ and $W$ are star triplets with non-identical ancestors.** Let $U$ and $W$ be two star triplets with two ancestor $r_u$ and $r_w$, respectively, such that $r_u \neq r_w$. As $U$ and $W$ are star triplets, $d_x^U$ and $d_y^W$ returns the distance from their corresponding ancestors $d_{xr_u}$ for all $x \in U$, and $d_{yr_w}$ for all $y \in W$, respectively. Now, according to the Claim 2, there exists a vertex triplet, say $\{u, w_1, w_2\}$ w.l.o.g., where $u \in U$ and $w_1, w_2 \in W$ such that $u$ is separated from $w_i$ for $i = 1, 2$ by both $r_u$ and $r_w$. Furthermore, the same $u$ identified above is separated from $r_w$ by $r_u$. This implies that $d_{uw_i} = d_{ur_u} + d_{r_u w_i} = d_{ur_u} + d_{r_u r_w} + d_{r_w w_i}$ for $i = 1, 2$. As we know that $r_u$ and $r_w$ are not identical, $d_{r_u r_w} \neq 0$, which implies that $d_{uw_i} \neq d_{ur_u} + d_{r_w w_i}$, where $i = 1, 2$. Thus we conclude the proof for the first configuration by showing that there exists a vertex $u \in U$ and a pair $w_1, w_2 \in W$ such that the identities in equation 2 and equation 3 do not hold.

**$U$ is a star triplet and $W$ is a non-star triplet in $G^{\text{J}}$.** We show that there exists a triplet $\{y, a, b\}$ where $y \in U$ and $a, b \in W$ such that identities in equation 2 and equation 3 do not hold. Let $W$ be a non-star triplet, and $U$ be a star triplet with the ancestor $r \in V$ in $G^{\text{J}}$. Now, as $U$ is a star triplet, $d_x^U$ returns the distance from its ancestor $d_{xr}$ for all $x \in U$. Suppose that there exists a vertex pair $x \in U$ and $a \in W$ for which $d_{xr} + d_a^W = d_{ax}$. We know that for a non-star triplet $W$, the computed distance $d_a^W \neq d_{ar}$ for any $a \in W$ from Assumption 4.1. Thus, for the pair $\{x, a\}$, $d_{xr} + d_{ar} \neq d_{xa}$. This implies from the Fact 1 that $x \not\perp\!\!\!\perp a \mid r$. Similarly, we can conclude that $x \not\perp\!\!\!\perp b \mid r$. Then, $y \perp\!\!\!\perp a \mid r$ and $y \perp\!\!\!\perp b \mid r$. Otherwise, one can construct a path between $y$ and $x$ that does not contain $r$ which violates the assumption that $U \ni x, y$ is a star triplet with ancestor $r$. As $y \perp\!\!\!\perp a \mid r$ and $y \perp\!\!\!\perp b \mid r$, using the Fact 1 we have that $d_{yr} + d_{ra} = d_{ya}$ and $d_{yr} + d_{rb} = d_{yb}$. As $a \in W$, and $d_{ar} \neq d_a^W$, thus, $d_{yr} + d_a^W \neq d_{ya}$. Similarly, for the pair $\{y, b\}$, we have that $d_{yr} + d_b^W \neq d_{yb}$. Thus, for the triplet $\{y, a, b\}$, the identities in Eq. equation 2 and equation 3 do not hold.

**$U$ and $W$ are both non-star triplets in $G^{\mathrm{J}}$.** The proof for this configuration follows from the Assumption 4.1.

Notice that these three cases combined proves that the TIA test returns TRUE if and only if the triplets considered are both start triplets that share a common ancestor. □

Now recall that the first phase of Subroutine 3 identifies the star triplets in $G^{\mathrm{J}}$, the observed ancestors in $G^{\mathrm{J}}$, and outputs a set $A_{\mathrm{hid}}$ such that $|A_{\mathrm{hid}}|$ equals to the number of hidden ancestors in $G^{\mathrm{J}}$. Formally, the result is as follows.

**Proposition B.8** (Correctness of Subroutine 3 in identifying ancestors)**.** *Given the pairwise distances $\{d_{ij}\}_{i,j\in V^{\mathrm{o}}}$, Subroutine 3 correctly identifies (a) the star triplets in $G^{\mathrm{J}}$, (b) the observed ancestors in $G^{\mathrm{J}}$, and (b) introduces a set $A_{hid}$ such that $|A_{hid}|$ equals to the number of hidden ancestors in $G^{\mathrm{J}}$*

*Proof.* Combining Lemma B.5 and Lemma B.7, we prove that the Subroutine 3 successfully cluster the star triplets in $G^{\mathrm{J}}$. Then, it partitions $\mathfrak{V}$ into $\mathfrak{V}_{\mathrm{obs}}$ and $\mathfrak{V}_{\mathrm{hid}}$ s.t. following is true: for any triplet collection $\mathcal{V}_i \in \mathfrak{V}_{\mathrm{obs}}$ ($\mathcal{V} \in \mathfrak{V}_{\mathrm{hid}}$ resp.), the ancestor of the triplets in $\mathcal{V}_i$ is observed (hidden resp.) Finally, Subroutine 3 outputs a set $A_{\mathrm{hid}}$ s.t. $|A_{\mathrm{hid}}| = |\mathfrak{V}_{\mathrm{hid}}|$. □

We show the correctness of Subroutine 3 in learning (a) $\{d_{ij}\}_{i^e\in V^{\mathrm{o}},j\in A_{\mathrm{hid}}}$, and (b) $\{d_{ij}\}_{i,j\in A_{\mathrm{hid}}}$.

**Claim 3.** *Fix any $a \in A^{\mathrm{J}}$, where $A^{\mathrm{J}}$ = set of all ancestors. Let $\mathcal{V}_a$ be the collection of vertex triplets which shares common ancestor. Then, any $i \in A^{\mathrm{J}}$ is s.t. at least one triplet in $\mathcal{V}_a$.*

*Proof.* Fix any ancestor $a \in A^{\mathrm{J}}$. Construct a triplet $T_i$ for a fixed vertex $i \neq a \in V^{\mathrm{J}}$ s.t. $a$ is the ancestor of $T_i$ in $G^{\mathrm{J}}$. From Lemma B.4: $a$ is a cut vertex in $G$. Thus, fixing $i$ and $a$, find another vertex $w \in V$ such that $a$ separates $i$ and $w$ in $G$. Hence, from Lemma B.3 we can conclude the following: $a$ is the ancestor for the triplet $T_i \triangleq \{i, a, w\}$ in $G^{\mathrm{J}}$. □

**Claim 4.** *Let $U_i$ and $U_j$ be both star triplets with ancestor $i$ and $j$ respectively, and $i \neq j$. Let $x \in U_i$ and $y \in U_j$ be a vertex pair such that $x \perp\!\!\!\perp y|i$ and $x \not\perp\!\!\!\perp y|j$ Then, $x \not\perp\!\!\!\perp i|j$.*

*Proof.* $x \perp\!\!\!\perp y|i$ implies any path between $x$ and $y$ contains $i$. $x \not\perp\!\!\!\perp y|j$ implies $\exists$ a path $\pi$ between $x$ and $y$ that does not contain $j$. Notice that, the path $\pi$ contains $i$. As $\pi$ contains both $x$ and $i$, $\exists$ a path between $x$ and $i$ which does not contain $j$. Hence, $x \not\perp\!\!\!\perp i|j$. □

**Lemma B.9.** *For any pair of distinct ancestors $i, j \in A^{\mathrm{J}}$, pick arbitrary triplets $U_i \in \mathcal{V}_i$ and $U_j \in \mathcal{V}_j$. Define the set $D(U_i, U_j)$ as follows:*

$$\Delta(U_i, U_j) \triangleq \left\{d_{xy} - (d_x^{U_i} + d_y^{U_j}) : x \in U_i, y \in U_j\right\} \tag{5}$$

*The most frequent element in $\Delta(U_i, U_j)$, that is,* $\mathrm{mode}(\Delta_{ij})$ *is the true distance $d_{ij}$ with respect to $G^{\mathrm{J}}$.*

*Proof.* To aid exposition, we suppose that $U_i = \{x_1, x_2, x_3\}$ and $U_j = \{y_1, y_2, y_3\}$. We also define for any $x \in U_i$ and $y \in U_j$: $\Delta(x, y) \triangleq d_{xy} - \left(d_x^{U_i} + d_y^{U_j}\right)$. Observe that according to the Claim 2, for two start triplets $U_i, U_j \in \binom{V^{\mathrm{o}}}{3}$ with non-identical ancestors, there exist a vertex, say $x_1 \in U_i$ and a pair, say $y_1, y_2 \in U_j$ such that following is true: $x_1 \perp\!\!\!\perp y_i|i$ and $x_1 \perp\!\!\!\perp y_i|j$ for $i = 1, 2$ Furthermore, the same $x_1$ (identified above) is separated from $j$ by $i$, that is, $x_1 \perp\!\!\!\perp j|i$. This similar characterization is also true for a vertex triplet where one vertex is from $U_j$ and a pair from $U_i$. Now observe that

$$\Delta(x_1, y_1) = d_{x_1y_1} - d_{x_1i} - d_{y_1j} = d_{x_1j} + d_{y_1j} - d_{x_1i} - d_{y_1j} = d_{x_1j} - d_{x_1i} = d_{ij}.$$

Similarly, it can be checked that $\Delta(x_1, y_2) = d_{ij}$. The similar calculation can be shown for the other triplet (where one vertex is from $U_j$ and a pair from $U_i$). In other words, we have demonstrated that 4 out of the 9 total distances in $D(U_i, U_j)$ are equal to $d_{ij}$. All that is left to be done is to show that no other value can have a multiplicity of four or greater.

Now, our main focus is to analyze the five remaining distances, i.e., $\Delta(x_3, y_3)$, $\Delta(x_3, y_1)$, $\Delta(x_3, y_2)$, $\Delta(x_1, y_3)$, and $\Delta(x_2, y_3)$, for two remaining configurations: (a) $x_3$ is separated from $y_3$ by only one vertex in $\{i, j\}$, and (b) $x_3 \not\perp\!\!\!\perp y_3|i$ *and* $x_3 \not\perp\!\!\!\perp y_3|j$. For configuration (a), consider without loss of generality that $x_3 \perp\!\!\!\perp y_3|i$ and $x_3 \not\perp\!\!\!\perp y_3|j$. Then, according to Claim 4, we have the following two possibilities:

1. $\underline{x_3 \perp\!\!\!\perp y_3|i, x_3 \not\perp\!\!\!\perp y_3|j, \text{ and } x_3 \perp\!\!\!\perp j|i}$: As $x_3 \not\perp\!\!\!\perp y_3|j$, it must be the case that $x_3 \perp\!\!\!\perp y_\nu|j$ for $\nu = 1, 2$. Otherwise, one can construct a path between $y_1$ and $y_\nu$ which does not contain $j$, and that violates the hypothesis that $U_j = \{y_1, y_2, y_3\}$ is a star triplet. Next, notice that in this set up, $x_3 \perp\!\!\!\perp j|i$. Now, notice the following:

$$\Delta(x_3, y_\nu) = d_{x_3 y_\nu} - d_{x_3 i} - d_{y_\nu j} = d_{x_3 j} + d_{y_\nu j} - d_{x_3 i} - d_{y_\nu j} = d_{x_3 i} + d_{ij} - d_{x_3 i} = d_{ij}.$$

   Therefore, for this set up, six distances are equal to $d_{ij}$.

2. $\underline{x_3 \perp\!\!\!\perp y_3|i, x_3 \not\perp\!\!\!\perp y_3|j, \text{ and } x_3 \not\perp\!\!\!\perp j|i}$: As $x_3 \not\perp\!\!\!\perp y_3|j$, it must be the case that $x_3 \perp\!\!\!\perp y_\nu|j$ for $\nu = 1, 2$. Otherwise, one can construct a path between $y_1$ and $y_\nu$ which does not contain $j$, and that violates the hypothesis that $U_j = \{y_1, y_2, y_3\}$ is a star triplet. $\Delta(x_3, y_\nu) = d_{x_3 y_\nu} - d_{x_3 i} - d_{y_\nu j} = d_{x_3 j} + d_{y_\nu j} - d_{x_3 i} - d_{y_\nu j} = d_{x_3 j} - d_{x_3 i}$. Now, $d_{x_3 j} - d_{x_3 i}$ equals to $d_{ij}$ implies that $x_3 \perp\!\!\!\perp j|i$ which contradicts the setup. Therefore, $\Delta(x_3, y_\nu)$ not equals to $d_{ij}$. Therefore, for this set up, even if three remaining distances are equal, correct $d_{ij}$ will be chosen.

In the following we will analyze the distance between $\Delta(x_3, y_3)$ and $\Delta(x_3, y_\nu)$ using the following assumption common in graphical models literature: For any vertex triplet $i, j, k \in \binom{V^o}{3}$, if $i \not\perp\!\!\!\perp j|k$, then $|d_{ij} - d_{ik} - d_{jk}| > \gamma$.

$$\begin{aligned}
&\Delta(x_3, y_3) - \Delta(x_3, y_\nu) = d_{x_3 y_3} - d_{x_3 i} - d_{y_3 j} - d_{x_3 y_\nu} + d_{x_3 i} + d_{y_\nu j}, \\
=&d_{x_3 i} + d_{y_3 i} - d_{x_3 i} - d_{y_3 j} - d_{x_3 j} - d_{y_\nu j} + d_{x_3 i} + d_{y_\nu j} = d_{y_3 i} + d_{x_3 i} - d_{y_3 j} - d_{x_3 j} = d_{x_3 y_3} - d_{y_3 j} - d_{x_3 j}.
\end{aligned}$$

Now, as $x_3 \not\perp\!\!\!\perp y_3|j$ according to Assumption 5.2, $|\Delta(x_3, y_3) - \Delta(x_3, y_\nu)| > \gamma$ for $\nu = 1, 2$. For configuration (b), we analyze the five remaining distances, i.e., $\Delta(x_3, y_3)$, $\Delta(x_3, y_1)$, $\Delta(x_3, y_2)$, $\Delta(x_1, y_3)$, and $\Delta(x_2, y_3)$, and show that these five distances can not be identical which in turn will prove the lemma.

$\underline{x_3 \not\perp\!\!\!\perp y_3|i \text{ and } x_3 \not\perp\!\!\!\perp y_3|j}$. For this configuration we note the following two observations:

O1 As $x_3 \not\perp\!\!\!\perp y_3|j$, it must be the case that $x_3 \perp\!\!\!\perp y_\nu|j$ for $\nu = 1, 2$. Otherwise, one can construct a path between $y_1$ and $y_\nu$ which does not contain $j$, and that violates the hypothesis that $U_j = \{y_1, y_2, y_3\}$ is a star triplet.

O2 Similarly, as $x_3 \not\perp\!\!\!\perp y_3|i$, it must be the case that $x_\nu \perp\!\!\!\perp y_3|i$ for $\nu = 1, 2$. Otherwise, one can construct a path between $x_1$ and $x_\nu$ which does not contain $i$, and that violates the hypothesis that $U_i = \{x_1, x_2, x_3\}$ is a star triplet.

Recall that our goal for configuration (b) is to analyze the distances $\Delta(x_3, y_3)$, $\Delta(x_3, y_1)$, $\Delta(x_3, y_2)$, $\Delta(x_1, y_3)$, and $\Delta(x_2, y_3)$. We start with the distance pair $\Delta(x_3, y_\nu)$ and $\Delta(x_3, y_3)$ for $\nu = 1, 2$.

$$\Delta(x_3, y_\nu) \overset{(a)}{=} d_{x_3 j} + d_{y_\nu j} - d_{x_3 i} - d_{y_\nu j} = d_{x_3 j} - d_{x_3 i},$$

where (a) follows from the O1. Furthermore, the distance $\Delta(x_3, y_3) = d_{x_3 y_3} - d_{y_3 j} - d_{x_3 i}$. Now, $\Delta(x_3, y_\nu)$ equals to $\Delta(x_3, y_3)$ implies that $d_{x_3 j} - d_{x_3 i} = d_{x_3 y_3} - d_{y_3 j} - d_{x_3 i}$ which is equivalent to saying that $d_{x_3 j} + d_{y_3 j}$ equals to $d_{x_3 y_3}$. Then, $d_{x_3 j} + d_{y_3 j} = d_{x_3 y_3}$ will imply $x_3 \perp\!\!\!\perp y_3|j$ – which contradicts the hypothesis of the configuration that $x_3 \not\perp\!\!\!\perp y_3|j$.

Thus, $\Delta(x_3, y_3)$ is not equal to $\Delta(x_3, y_\nu)$ for $\nu = 1, 2$. (based on O1). Similarly, (based on the O2) $\Delta(x_3, y_3)$ is not equal to $\Delta(x_\nu, y_3)$ for $\nu = 1, 2$. Thus, the distance $\Delta(x_3, y_3)$ is not equal to any of the following distances: $\Delta(x_3, y_1), \Delta(x_3, y_2), \Delta(x_1, y_3)$, and $\Delta(x_2, y_3)$.

Now all that remains to prove the lemma is to show that the 4 (remaining) distances $\Delta(x_3, y_1)$, $\Delta(x_3, y_2)$, $\Delta(x_1, y_3)$, and $\Delta(x_2, y_3)$ are not identical. To this end, we analyze two distances: $\Delta(x_1, y_3)$ and $\Delta(x_3, y_1)$. First notice from the O2 that $\Delta(x_1, y_3) = d_{x_1 i} + d_{x_3 i} - d_{x_3 i} - d_{y_3 j}$ equals to $d_{y_3 i} - d_{y_3 j}$, and $\Delta(x_3, y_1) = d_{x_3 j} + d_{y_3 j} - d_{x_3 i} - d_{y_3 j}$ equals to $d_{x_3 j} - d_{x_3 i}$. As neither $i$ nor $j$ is separating $x_3$ from $y_3$, the event that $d_{y_3 i} - d_{y_3 j}$ equals to $d_{x_3 j} - d_{x_3 i}$ happens only on a set of measure zero. We end this proof by computing the distance between $\Delta(x_3, y_3)$ and $\Delta(x_3, y_\nu$.

$$\Delta(x_3, y_3) - \Delta(x_3, y_\nu) = d_{x_3 y_3} - d_{y_3 j} - d_{x_3 i} - d_{x_3 j} - d_{y_\nu j} + d_{x_3 i} + d_{y_\nu j} = d_{x_3 y_3} - d_{y_3 j} - d_{x_3 j}.$$

Now, as $x_3 \not\perp\!\!\!\perp y_3 | j$, according to Assumption 5.2, $|\Delta(x_3, y_3) - \Delta(x_3, y_\nu)| > \gamma$ for $\nu = 1, 2$. □

**Lemma B.10** (Correctness in *extending the distances.*)**.** *Given $\{d_{ij}\}_{i,j \in V^{\text{o}}}$, Subroutine 3 correctly learns (a) $\{d_{ij}\}_{i,j \in V^{\text{o}} \cup A}$ and (b) $\{d_{ij}\}_{i,j \in A_{hid}}$, where $A \triangleq A_{obs} \cup A_{hid}$.*

*Proof.* Follows directly from Claim 3 and Lemma B.9. □

**Lemma B.11** (Correctness in learning clusters)**.** *Subroutine 4 correctly learns leaf clusters and internal clusters.*

*Proof.* As the distances $\{d_{ij}\}_{i,j \in A_{\text{hid}}}$ and $\{d_{ij}\}_{i,j \in V^{\text{o}} \cup A_{\text{hid}}}$ are learned correctly by Subroutine 3, where $V^{\text{o}}$ and $A_{\text{hid}}$ is the set of observed vertices, and hidden ancestors, respectively, the correctness of learning the leaf clusters and internal clusters follows from Fact 1. □

**Lemma B.12.** *Let $L \subset 2^V$ be a subset of vertices in $G$ s.t. only noisy samples are observed from the vertices in $L$. Then, $\exists$ a vertex $v \in L$, where $v \in V_{\text{cut}}$, s.t. $v$ separates $L \setminus \{v\}$ from the remaining vertices $v' \in V_{\text{cut}} \setminus \{v\}$ Let $L^e$ be the noisy counterpart of $L$. The noiseless counterpart of $x^e$ is a non-cut vertex if and only if $\exists$ at least a pair $y^e, z^e \in L^e \setminus \{x^e\}$ such that $TIA(\{x^e, y^e, \alpha_1^e\}, \{x^e, z^e, \alpha_2^e\})$ returns* False*, where $\alpha_1^e, \alpha_2^e \in V^{\text{J}} \setminus L^e$.*

*Proof.* ($\Rightarrow$) Suppose that $x$ is a non-cut vertex of a non-trivial block in $G$. We show the existence of a pair $y^e, z^e \in L^e \setminus \{x^e\}$ in $V^{\text{J}}$ such that $TIA(\{x^e, y^e, \alpha_1\}, \{x^e, z^e, \alpha_2^e\})$ returns False, where $\alpha_1^e, \alpha_2^e \in V^{\text{J}} \setminus L^e$. From Section 3 we have that any non-trivial block in $G$ has at least three vertices. Hence, there exists another vertex $y^e \in L^e$ for which the noiseless counterpart is a non-cut vertex. We will show that one of $\{x^e, y^e, \alpha_1^e\}$ and $\{x^e, z^e, \alpha_2^e\}$ is not a star triplet, where $z^e \in L^e \setminus x^e, y^e$, and $\alpha_1^e, \alpha_2^e \in V^{\text{J}} \setminus L^e$. Then, the $TIA(\{x^e, y^e, \alpha_1^e\}, \{x^e, z^e, \alpha_2^e\})$ being False will follow from Lemma B.7. As $x$ and $y$ both are non-cut vertices, there does not exist a cut vertex that separates $x$ and $y$ in $G$, which implies that there does not exist an ancestor $a$ in $G^{\text{J}}$ s.t. $x^e \perp\!\!\!\perp y^e | a$. Hence, $\{x^e, y^e, \alpha_1^e\}$ is not a star triplet in $G^{\text{J}}$. Then, the proof follows from Lemma B.7.

($\Leftarrow$ ) Notice that the pair $(\{x^e, y^e, \alpha_1^e\}, \{x^e, z^e, \alpha_2^e\})$ can not share an ancestor, as it would violate the claim that $\{x^e, y^e, z^e\}$ is in a leaf cluster. Then, from Lemma B.7 we have that if $TIA(\{x^e, y^e, \alpha_1^e\}, \{x^e, z^e, \alpha_2^e\})$ returns False, then at least one of the triplets is a non-star triplet, which rules out the existence of star triplets with non-identical ancestor. Suppose that $\{x^e, y^e, \alpha_1^e\}$ is a non-star triplet. As $\alpha_1^e \notin L^e$, an ancestor separates $x^e$ and $\alpha_1^e$, and $y^e$ and $\alpha_1^e$. Then, the ancestor identified above does not separate $x^e$ and $y^e$. Hence in $G$, there does not exist a cut vertex that separates $x$ and $y$. □

Unidentifiability of the articulation point from a leaf cluster. According to Lemma B.12 the NonCutTest returns the non-cut vertices of a non-trivial block from a leaf cluster, and the next (immediate) step is to learn the cut vertices of the non-trivial blocks. We now present a claim which shows a case where identifying the articulation point from a leaf cluster is not possible. This ambiguity is exactly the ambiguity (in robust model selection problem) of the *label swapping of the leaf vertices with their neighboring internal vertices* of a tree-structured Gaussian graphical models Katiyar et al. (2019).

**Claim 5.** *Let (i) a vertex $v \in V_{\text{cut}}$ separates a subset $L \subset 2^V$ of vertices from any $v' \in V_{\text{cut}} \setminus v$ where $L$ contains at least one leaf vertex, and (ii) $L^e$ be the noisy counterpart of $L$. Then, there exist at least two vertices $x_1^e, x_2^e \in L^e \cup \{v^e\}$ such that $TIA(\{x^e, y^e, \alpha_1^e\}, \{x^e, z^e, \alpha_2^e\})$ returns* TRUE *for any pair $y^e, z^e \in L^e \cup \{v^e\}$ where $x^e \in \{x_1^e, x_2^e\}$, and $\alpha_1^e, \alpha_2^e \in V^{\text{J}} \setminus L^e \cup \{v^e\}$.*

*Proof.* As $v$ is a cut vertex, $v^e \perp\!\!\!\perp y^e | v$, $v^e \perp\!\!\!\perp \alpha_1^e | v$, and $y^e \perp\!\!\!\perp \alpha_1^e | v$. Here, $v$ is a unique separator since no other cut vertex (or ancestor in $G^{\text{J}}$) separates $v^e$ and $y^e$. Hence, $v$ is the ancestor of $\{v^e, y^e, \alpha_1^e\}$ in $G^{\text{J}}$. Similarly, one can construct another triplet $\{v^e, x^e, \alpha_1^e\}$ which has an ancestor $v$ in $G^{\text{J}}$. Hence, $TIA(\{v^e, y^e, \alpha_1^e\}, \{v^e, x^e, \alpha_2^e\})$ will return TRUE. Now, let us consider a leaf vertex $x_1$ in $L$. Now, $x_1^e \perp\!\!\!\perp \alpha_1^e | v$, $x_1^e \perp\!\!\!\perp y^e | v$, and $\alpha_1^e \perp\!\!\!\perp y^e | v$. Hence, $v$ is the ancestor of $\{x_1^e, y^e, \alpha_1^e\}$. Similarly, one can construct another triplet such that $v$ is the ancestor of $\{x_1^e, x^e, \alpha_2^e\}$. Hence, $TIA(\{x_1^e, y^e, \alpha_1^e\}, \{x_1^e, x^e, \alpha_2^e\})$ will return TRUE. □

**Proposition B.13.** *Suppose that Subroutine 6 is invoked with the correct leaf clusters and internal clusters. Further suppose that* NONCUTTEST *succeeds in identifying the non-cut vertices of a non-trivial block. Then, Subroutine 6 correctly learns $\mathcal{P}_{\text{op}}$ and $A_{\text{op}}$ for $\mathcal{T}_{\text{op}}$.*

*Proof.* According to Lemma B.12, Subroutine 6 correctly learns the non-cut vertices of any non-trivial block $I$ with more than one cut vertices. If the cut vertex is observed, then it is identified in Subroutine 3, and declared as one the articulation points of the vertex $I$ in $\mathcal{P}_{\text{op}}$. Otherwise, the noisy counterpart belongs to a leaf cluster associated with an hidden ancestor, and the cut vertex be identified by selecting the label of the leaf cluster which is associated with the hidden ancestor (unobserved cut vertex of non-trivial block.)

□

We now establish the correctness of NoMAD in the learning the edge set $E_{\text{op}}$ for $\mathcal{T}_{\text{op}}$. This goal is achieved correctly by Procedure NONBLOCKNEIGHBORS of NoMAD.

**Proposition B.14.** *Suppose that Procedure 4 is invoked with the correct $\mathcal{P}_{\text{op}}$ and $A_{\text{op}}$. Then, Procedure 4 returns the edge set $E_{\text{op}}$ correctly.*

*Proof.* Procedure NONBLOCKNEIGHBORS correctly learns the neighbors of any fixed articulation point in $A_{\text{op}}$ by ruling out the non-neighbor articulation points in $\mathcal{T}_{\text{op}}$. First, the procedure gets rid of the articulation points which are separated from the articulation points of the same vertex in $A_{\text{op}}$. Then, from the remaining articulation points it chooses the set of all those articulation points such that no pair in the set is separated from each other by the fixed articulation point. Then, Procedure 4 creates edges between vertices which contains the neighboring articulation points. □

**Constructing the Equivalence Class.** Finally, in order to show that we can construct the equivalence class $[G]$ from the articulated set tree $\mathcal{T}_{\text{op}}$, we note some additional definitions in the following. For graph $G$, let $B_{\text{non-cut}}$ be the set of all non-cut vertices in a non-trivial block $B$. Define $\mathcal{B}_{\text{non-cut}} \triangleq \bigcup_{B \in \mathcal{B}^{\text{NT}}} B_{\text{non-cut}}$, where $\mathcal{B}^{\text{NT}}$ is the set of all non-trivial blocks. Let a set $F_i$ referred as a *family* be defined as $\{v : \deg(v) = 1 \text{ and } \{v, i\} \in E(G)\} \cup \{i\}$ where $E(G)$ is the edge set of $G$, and let $\mathcal{F} = \bigcup_{i \in V} F_i$. Let $K$ be the set of cut vertices whose neighbors do not contain a leaf vertex in $G$. For any vertex $k \in K$, let a family $F_k \in \mathcal{F}$ be such that there exists a vertex $f \in F_k$ such that $\{k, f\} \in E(G)$. For example, in Fig. 1a, $\mathcal{F} = \{\{10, 11, 12, 13\}, \{14, 15, 16\}, \{17, 20, 21\}\}$; two sets $\{1, 2, 3\}, \{18, 19\}$ in $\mathcal{B}_{\text{non-cut}}$, and $K = \{4, 6, 7, 8, 9\}$. Also, for example, $F_4 = \{10, 11, 12, 13\}$. For any arbitrary graph $\widetilde{G}$, let $B_{\text{non-cut}}(\widetilde{G})$, $\mathcal{F}(\widetilde{G})$, and $K(\widetilde{G})$ be the corresponding sets from $\widetilde{G}$.

Now, notice that in $\mathcal{T}_{\text{op}}$, each vertex $k \in K$ has at least an edge in $\mathcal{T}_{\text{op}}$. Let $N_{\text{art}}(k)$ be the neighbors of $k \in K$ in the edge set $E_{\text{op}}$ returned for $\mathcal{T}_{\text{op}}$. Now, notice that as as long as $\mathcal{B}_{\text{non-cut}}$, $\mathcal{F}$, and $K$ are identified correctly in $\mathcal{T}_{\text{op}}$, and the following condition holds in $E_{\text{op}}$ for any $i \in N_{\text{art}}(k)$ for each $k \in K$: (a) if $i \in K$, then $\{i, k\} \in E(G)$, and (b) otherwise, there exists a vertex $j \in F_k$ such that $\{j, k\} \in E(G)$. Informally, identifying $\mathcal{B}_{\text{non-cut}}$ and $\mathcal{F}$ correctly, makes sure that vertices that constructs the local neighborhoods of any graph in $[G]$ are identical; identifying $K$ correctly, and satisfying the above-mentioned condition makes sure that the correct articulation points are recovered. Notice that the sets $\mathcal{B}_{\text{non-cut}}$, $\mathcal{F}$, and $K$ are identical in all

the graphs in Fig. 2. Following proposition shows that the sets $\mathcal{B}_{\text{non-cut}}$, $\mathcal{F}$, and $K$ are identified correctly from $\mathcal{T}_{\text{op}}$.

**Lemma B.15.** *Let $\widetilde{G}$ be an arbitrary graph. Then, $\widetilde{G} \in [G]$ if and only if the following holds:*

1. *$\mathcal{B}_{non\text{-}cut}(\widetilde{G}) = \mathcal{B}_{non\text{-}cut}(G)$, $\mathcal{F}(\widetilde{G}) = \mathcal{F}(G)$, and $K(\widetilde{G}) = K(G)$.*
2. *For any vertex $k \in K$, let a family $F_k \in \mathcal{F}$ be such that there exists a vertex $f \in F_k$ such that $\{k, f\} \in E(\widetilde{G})$. Now, for any neighbor $i \in N(k)$: (a) if $i \in K$, then $\{i, k\} \in E(\widetilde{G})$, and (b) otherwise, there exists a vertex $j \in F_k$ such that $\{j, k\} \in E(\widetilde{G})$.*

*Proof.* ($\Rightarrow$) The forward implication follows from Definition 3.2.

($\Leftarrow$) For the reverse implication, notice that the first condition is associated with the equality between sets. $\mathcal{B}_{\text{non-cut}}(\widetilde{G}) = \mathcal{B}_{\text{non-cut}}(G)$ implies non-cut vertices are identified correctly, and $\mathcal{F}(\widetilde{G}) = \mathcal{F}(G)$ implies families are identified correctly. The second condition implies that an edge associated with a vertex $k \in K$ will have an ambiguity when the other vertex is from a family. Recall that from Definition 3.2, the label of a cut vertex can be swapped with it's neighbor leaf vertices. □

The reverse implication of the above-mentioned proof can be understood as follows: Identifying $\mathcal{B}_{\text{non-cut}}$ and $\mathcal{F}$ ensures that essentially the *local structures* are identical between $G$ and $\widetilde{G}$. Recovering $K$ correctly and satisfying the second condition ensure that these local structures are correctly attached at the appropriate points.

**Proposition B.16** (Correctness in Learning the Equivalence Class)**.** *Suppose that $\mathcal{P}_{\text{op}}$, $A_{\text{op}}$, and $E_{\text{op}}$ returned by $\mathcal{T}_{\text{op}}$ is correct. Then, following is true: (a)The sets $\mathcal{B}_{non\text{-}cut}$, $\mathcal{F}$, and $K$ are identified correctly, and (b) the condition is true for $N_{art}(k)$ for each $k \in K$.*

*Proof.* We first show that NoMAD correctly identifies the sets $\mathcal{B}_{\text{non-cut}}$, $\mathcal{F}$, and $K$. By Lemma B.12, Subroutine 6 correctly identifies the set $\mathcal{B}_{\text{non-cut}}$. Now, recall that each $F \in \mathcal{F}$ is a set of vertices constructed with a cut vertex and its neighbor leaf vertices. Hence, each family $F \in \mathcal{F}$ is captured in one of the leaf clusters returned by Subroutine 4. As Subroutine 6 correctly identifies the non-cut vertices from each leaf cluster, $\mathcal{F}$ is identified correctly. Finally, by Claim 5, the ambiguity in learning an articulation point is present only when a cut vertex has leaf vertex as it's neighbor; but $K$ does not contain such cut vertices. Hence, Subroutine 6 correctly learns $K$. We now show that above-mentioned condition is satisfied for the neighbor articulation points in $N_{\text{art}}(k)$ for any $k \in K$. As $K$ are identified correctly by Subroutine 6, and the Procedure 4 returns correct $N_{\text{art}}(k)$, it is clear that if any neighbor articulation point $i \in N_{\text{art}}(k) \cap K$, then $\{i, k\} \in E(G)$. Now, suppose that $i \notin N_{\text{art}}(k) \cap K$. Then, from Definition 3.2, the label of a cut vertex can be swapped with it's neighbor leaf vertices. As each family $F \in \mathcal{F}$ are identified correctly, there exists a vertex $j \in F_k$ (which is an *unidentified* cut vertex in $G$) such that $\{i, j\} \in E(G)$. □

## C Sample Complexity Result

Recall that NoMAD returns the equivalence class of a graph $G$ while having access only to the noisy samples according to the problem setup in Section 3.1. But, in the finite sample regime, instead of the population quantities, we only have access to samples. We will use these to create natural estimates $\widehat{\rho}_{ij}$, for all $i, j \in V^{\text{O}}$ of the correlation coefficients given by $\widehat{\rho}_{ij} \triangleq \frac{\widehat{\Sigma^o}_{ij}}{\sqrt{\widehat{\Sigma^o}_{ii}\widehat{\Sigma^o}_{jj}}}$, where $\widehat{\Sigma^o}_{ij} = \frac{1}{n}\sum_{k=1}^{n} y_i^{(k)} y_j^{(k)}$. Indeed, these are random quantities and therefore we need to make slight modifications to the algorithm as follows:

**Change in the TIA test.** We start with the following assumption: For any triplet pair $U, W \in \binom{V}{3} \setminus \mathcal{V}_{\text{star}} \cup \mathcal{V}_{\text{sep}}$ and any vertex pair $(x, a) \in U \times W$, there exists a constant $\zeta > 0$, such that $\left|d_x^U + d_a^W - d_{xa}\right| > \zeta$. As we showed in Lemma B.7, for any pair $U, W \in \binom{V}{3} \setminus \mathcal{V}_{\text{star}} \cup \mathcal{V}_{\text{sep}}$, there exists at least one triplet $\{x, a, b\}$ where $x \in U$ and $a, b \in W$ such that $d_{xa} - d_x^U - d_a^W \neq 0$ and $d_{xb} - d_x^U - d_b^W \neq 0$. Hence, the observation

in Lemma B.7 motivates us to replace the exact equality testing in the TIA test in Definition 3.5 with the following hypothesis test against zero: $\max\left\{\left|\widehat{d}_{xa}-\widehat{d}^U_x-\widehat{d}^W_a\right|,\left|\widehat{d}_{xb}-\widehat{d}^U_x-\widehat{d}^W_b\right|\right\}\leq\xi$, for some $\xi<\frac{\zeta}{2}$.

**Change in the Mode test.** In order to compute the distance between the hidden ancestors in the finite sample regime, we first recall from (the proof of) Lemma B.9 that there are at least 4 instances (w.l.o.g.) $\Delta(x_1,y_1),\Delta(x_1,y_2),\Delta(x_2,y_1)$, and $\Delta(x_2,y_2)$ where $\Delta(x,y)$ where $x\in U_i$ and $y\in U_j$ such that equals to $d_{ij}$. We also showed that no set of identical but incorrect distance has cardnilaity more than two. Hence, In the finite sample regime, we replace the mode test in Subroutine 3 with a more robust version, which we call the $\epsilon_d-$mode test, where $\epsilon_d<\min(\frac{\xi}{14},\gamma)$ based on the following definition.

**Definition C.1** ($\epsilon_d-$mode)**.** *Given a set of real numbers $\{r_1,\ldots,r_n\}$, let $S_1,\ldots,S_k$ be a partition where each $r,r'\in S_i$ is such that $|r-r'|<\epsilon_d$ for each $i$. Then, the $\epsilon_d$-mode of the this set is defined as selecting an arbitrary number from the partition with the largest cardinality.*

In the finite sample regime, we run NoMAD with the mode replaced by the $\epsilon_d$-mode defined above such that $\epsilon_d<\min(\frac{\xi}{14},\gamma)$. We will call this modified mode test as the $\epsilon_d$-mode test.

**Change in Separation test.** For any triplet $(i,j,k)\in\binom{V^{\text{O}}}{3}$, in order to check whether $i\perp\!\!\!\perp j|k$, instead of the equality test in Fact 1, we modified the test for the finite sample regime as follows: $|\widehat{d}_{ij}-\widehat{d}_{ik}-\widehat{d}_{jk}|<\frac{\epsilon_d}{6}$. We now introduce two new notations to state our main result. Let $\rho_{\min}(p)=\min_{i,j\in\binom{p}{2}}|\rho_{ij}|$ and $\kappa(p)=\log((16+(\rho_{\min}(p))^2\,\epsilon_d^2)/(16-(\rho_{\min}(p))^2\,\epsilon_d^2))$, where $\epsilon_d=\min(\frac{\xi}{14},\gamma)$, where $\gamma$ is from Assumption 5.2.

**Theorem C.3.** *Suppose the underlying graph $G$ of a faithful GGM satisfies Assumptions 5.2-5.3. Fix any $\tau\in(0,1]$. Then, there exists a constant $C>0$ such that if the number of samples $n$ satisfies $n>C\left(\frac{1}{\kappa(p)}\right)\max\left(\log\left(\frac{p^2}{\tau}\right),\log\left(\frac{1}{\kappa(p)}\right)\right)$, then with probability at least $1-\tau$, NoMAD accepting $\widehat{d}_{ij}$ outputs the equivalence class $[G]$.*

*Proof.* First, there are (at most) seven pairwise distances to be estimated in terms of $\max\left\{\left|\widehat{d}_{xa}-\widehat{d}^U_x-\widehat{d}^W_a\right|,\left|\widehat{d}_{xb}-\widehat{d}^U_x-\widehat{d}^W_b\right|\right\}$. Therefore, the probability that our algorithm fails is bounded above by the probability that there exists a pairwise distance estimate that is $\xi/14$ away from its mean. To this end, let us denote a bad event $B_{i,j}$ for any pair $i,j\in V^{\text{O}}$ as the following:

$$B_{i,j}\triangleq\{|d_{ij}-\widehat{d}_{ij}|\geq\epsilon_d\}. \tag{6}$$

Then, the error probability $\mathbb{P}[[\mathcal{T}_{\text{algo}}]\neq[G]]$ is upper bounded as

$$\mathbb{P}\left([\mathcal{T}_{\text{algo}}]\neq[G]\right)\leq\mathbb{P}\left(\bigcup_{i,j\in V^{\text{O}}}B_{i,j}\right)\leq\sum_{i,j\in V^{\text{O}}}\mathbb{P}\left(B_{i,j}\right), \tag{7}$$

where $[\mathcal{T}_{\text{algo}}]$ is the output equivalence class. We now consider two following events: $K_{i,j}\triangleq\{|\widehat{\rho}_{ij}|\leq\frac{\rho_{\min}}{2}\}$ [6], and $R_{i,j}\triangleq\{|\rho_{ij}-\widehat{\rho}_{ij}|<\frac{\rho_{\min}\epsilon_d}{2}\}$. We will upper bound $\mathbb{P}(B_{i,j})$ for any pair $i,j$ using $\mathbb{P}(K_{i,j})$ and $\mathbb{P}(R_{i,j})$. Before that, notice the following chain of implications:

$\left(|\rho_{ij}-\widehat{\rho}_{ij}|<\frac{\rho_{\min}\times\epsilon_d}{2}\right)\Rightarrow\left(||\rho_{ij}|-|\widehat{\rho}_{ij}||<\frac{\rho_{\min}\times\epsilon_d}{2}\right)\Rightarrow\left(\left|d_{ij}-\widehat{d}_{ij}\right|<\frac{\left||\rho_{ij}|-|\widehat{\rho}_{ij}|\right|}{\min\left(|\widehat{\rho}_{ij}|,|\rho_{ij}|\right)}\right)\Rightarrow$
$\left(\left|d_{ij}-\widehat{d}_{ij}\right|<\frac{\left||\rho_{ij}|-|\widehat{\rho}_{ij}|\right|}{\min\left(\frac{\rho_{\min}}{2},\rho_{\min}\right)}\right)\Rightarrow\left(\left|d_{ij}-\widehat{d}_{ij}\right|<\frac{\frac{\rho_{\min}}{2}\times\epsilon_d}{\frac{\rho_{\min}}{2}}\right)\Rightarrow\left(\left|d_{ij}-\widehat{d}_{ij}\right|<\epsilon_d\right)$. These implications establish that $R_{i,j}\cap K^c_{i,j}\subseteq B^c_{i,j}$. Notice that as $R_{i,j}\cap K^c_{i,j}\subseteq B^c_{i,j}\cap K^c_{i,j}$, it will imply that $\mathbb{P}(B^c_{i,j}\cap K^c_{i,j})\geq\mathbb{P}(R_{i,j}\cap K^c_{i,j})$. Now, we can write the following bound:

$$\mathbb{P}(B_{i,j}|K^c_{i,j})\leq\mathbb{P}(R^c_{i,j}|K^c_{i,j}). \tag{8}$$

[6] for notational clarity we write $\rho_{\min}$ instead of $\rho_{\min}(p)$

Then, $\mathbb{P}(B_{ij})$ can be upper bounded as follows:

$$\mathbb{P}\left(B_{i,j}\right) = \mathbb{P}\left(B_{i,j}|K_{i,j}\right)\mathbb{P}\left(K_{i,j}\right) + \mathbb{P}\left(B_{i,j}|K^c_{i,j}\right)\mathbb{P}\left(K^c_{i,j}\right), \tag{9}$$

$$\leq \mathbb{P}\left(B_{i,j}|K_{i,j}\right)\mathbb{P}\left(K_{i,j}\right) + \mathbb{P}\left(R^c_{i,j}|K^c_{i,j}\right)\mathbb{P}\left(K^c_{i,j}\right), \tag{10}$$

$$\leq \left(1 \times \mathbb{P}\left(K_{i,j}\right)\right) + \left(\mathbb{P}\left(R^c_{i,j}|K^c_{i,j}\right) \times 1\right). \tag{11}$$

Then, $\mathbb{P}\left([\mathcal{T}_{\text{algo}}] \neq [G]\right)$ can be further bounded as

$$\mathbb{P}\left([\mathcal{T}_{\text{algo}}] \neq [G]\right) \leq \sum_{i,j \in V^{\text{o}}} \mathbb{P}\left(B_{i,j}\right) \leq \sum_{i,j \in V^{\text{o}}} \mathbb{P}\left(K_{i,j}\right) + \sum_{i,j \in V^{\text{o}}} \mathbb{P}\left(R^c_{i,j}|K^c_{i,j}\right).$$

Because $\mathbb{P}\left(R^c_{i,j}|K^c_{i,j}\right) < \mathbb{P}\left(R^c_{i,j}\right)/\mathbb{P}\left(K^c_{i,j}\right)$, we note that

$$\mathbb{P}\left([\mathcal{T}_{\text{algo}}] \neq [G]\right) \leq \sum_{i,j \in V^{\text{o}}} \mathbb{P}\left(K_{i,j}\right) + \sum_{i,j \in V^{\text{o}}} \frac{\mathbb{P}\left(R^c_{i,j}\right)}{\mathbb{P}\left(K^c_{i,j}\right)}.$$

We now find the required number of samples $n$ in order for $\mathbb{P}\left([\mathcal{T}_{\text{algo}}] \neq [G]\right)$ to be bounded by $\tau$. Before computing $n$ we note an important inequality from Kalisch & Bühlman (2007) which we use in bounding all the following events. For any $0 < \epsilon \leq 2$, and $\sup_{i \neq j} |\rho_{ij}| \leq M < 1$, following is true.

$$\mathbb{P}\left(|\widehat{\rho}_{ij} - \rho_{ij}| > \epsilon\right) \leq C_\rho \left(n-2\right) \exp\left(-\left(n-4\right)\log\left(\frac{4+\epsilon^2}{4-\epsilon^2}\right)\right), \tag{12}$$

for some constant $0 < C_\rho < \infty$ depending on $M$ only.

We now note the following assumption on bounded correlation which is a common assumption in learning the graphical models: $0 < \rho_{\min} \leq \rho_{\max} < 1$. Now notice that, $\left(|\widehat{\rho}_{ij}| \leq \frac{\rho_{\min}}{2}\right)$ together with $|\rho_{ij}| \geq \rho_{\min}$ implies that $|\rho_{ij}| - |\widehat{\rho}_{ij}| \geq \rho_{\min} - \frac{\rho_{\min}}{2} = \frac{\rho_{\min}}{2}$, since $\rho_{\min} > \frac{\rho_{\min}}{2}$. Furthermore, $|\rho_{ij} - \widehat{\rho}_{ij}| \geq |\rho_{ij}| - |\widehat{\rho}_{ij}|$ implies that $|\rho_{ij} - \widehat{\rho}_{ij}| \geq \frac{\rho_{\min}}{2}$. Then, we have the following:

$$\mathbb{P}\left(K_{i,j}\right) \leq \mathbb{P}\left(|\rho_{ij} - \widehat{\rho}_{ij}| \geq \frac{\rho_{\min}}{2}\right) \leq C_\rho \left(n-2\right) \exp\left(-\left(n-4\right)\log\left(\frac{16+\rho^2_{\min}}{16-\rho^2_{\min}}\right)\right). \tag{13}$$

Eq. equation 13 follows from Eq. equation 12. Now, According to Claim 6,

$$n_1 > \max\left(C_1 \frac{\log\left(\frac{2C_\rho\binom{p}{2}}{\tau}\right)}{\log\left(\frac{16+\rho^2_{\min}}{16-\rho^2_{\min}}\right)} \times \frac{C_2 C_1}{(C_1-1)\log\left(\frac{16+\rho^2_{\min}}{16-\rho^2_{\min}}\right)}, \log\left(\frac{\mathrm{C}_1}{(\mathrm{C}_1-1)\log\left(\frac{16+\rho^2_{\min}}{16-\rho^2_{\min}}\right)}\right)\right) + 4 \tag{14}$$

implies $\sum_{i,j \in V^{\text{o}}} \mathbb{P}(K_{i,j}) < \frac{\tau}{2}$,

$$n_3 > \max\left(C_1 \frac{\log\left(\frac{C_\rho}{1-\tau'}\right)}{\log\left(\frac{16+\rho^2_{\min}}{16-\rho^2_{\min}}\right)}, \frac{C_2 C_1}{(C_1-1)\log\left(\frac{16+\rho^2_{\min}}{16-\rho^2_{\min}}\right)} \times \log\left(\frac{\mathrm{C}_1}{(\mathrm{C}_1-1)\log\left(\frac{16+\rho^2_{\min}}{16-\rho^2_{\min}}\right)}\right)\right) + 4 \tag{15}$$

implies $\mathbb{P}(K^c_{i,j}) > \tau'$, where $\tau' > 1 - C_\rho$, and

$$n_4 > \max\left(C_1 \frac{\log\left(\frac{2C_\rho\binom{p}{2}}{\tau\tau'}\right)}{\log\left(\frac{16+\rho^2_{\min}\epsilon^2_{\mathrm{d}}}{16-\rho^2_{\min}\epsilon^2_{\mathrm{d}}}\right)}, \frac{C_2 C_1}{(C_1-1)\log\left(\frac{16+\rho^2_{\min}\epsilon^2_{d}}{16-\rho^2_{\min}\epsilon^2_{d}}\right)} \times \log\left(\frac{\mathrm{C}_1}{(\mathrm{C}_1-1)\log\left(\frac{16+\rho^2_{\min}\epsilon^2_{\mathrm{d}}}{16-\rho^2_{\min}\epsilon^2_{\mathrm{d}}}\right)}\right)\right) + 4 \tag{16}$$

implies $\mathbb{P}(R^c_{i,j}) < \frac{\tau\tau'}{2\binom{p}{2}}$. Now, notice that $n_2 \triangleq \max(n_3, n_4)$ implies $\frac{\mathbb{P}(R^c_{i,j})}{\mathbb{P}(K^c_{i,j})} < \frac{\tau}{2\binom{p}{2}}$. Therefore, acquiring at least $n_2$ samples will imply $\sum_{i,j \in V^o} \frac{\mathbb{P}(R^c_{i,j})}{\mathbb{P}(K^c_{i,j})} < \frac{\tau}{2}$. Finally, for $\mathbb{P}([\mathcal{T}_{\text{algo}}] \neq [G])$ to be upper bounded by $\tau$, it is sufficient for the number of samples $n$ to satisfy $n > \max(n_1, n_2)$. □

**Claim 6.** *There exist positive constants $T, C$, and $\widetilde{\alpha}$ such that if $n > \max(T, C \times \widetilde{\alpha}\log\widetilde{\alpha})$, then $n - \widetilde{\alpha}\log(n) > T$.*

*Proof.* We start the proof with the following claim: Suppose that there exists a constant $C_1, C_2$ where $C_1 < C_2$ such that $C_1 m\log m < n < C_2 m\log m$. Notice that for $m$ sufficiently large ($m > C_2$), we can show that $n > m\log n$. Therefore, for some constant $C_1, C_2$, $n > C_2 \times \frac{C_1}{(C_1-1)\alpha}\log\left(\frac{C_1}{(C_1-1)\alpha}\right)$ implies $n > \frac{C_1}{(C_1-1)\alpha}\log(n)$. Now, suppose that $\max\left(C_1T, \frac{C_2C_1}{(C_1-\alpha)}\log\left(\frac{C_1}{(C_1-1)\alpha}\right)\right) = C_1T$. Then, $n > C_1T$ implies $n > C_2 \times \frac{C_1}{(C_1-1)\alpha}\log\left(\frac{C_1}{(C_1-1)\alpha}\right)$. Then, from the initial claim we have that $n > \frac{C_1}{(C_1-1)\alpha}\log(n)$. Then, $n\frac{(C_1-1)}{C_1} > \frac{1}{\alpha}\log(n)$, and $n - \frac{1}{\alpha}\log(n) > \frac{n}{C_1}$. As $\frac{n}{C_1} > T$, we have that $n - \frac{1}{\alpha}\log(n) > T$. Further, suppose that $\max\left(C_1T, \frac{C_2C_1}{(C_1-\alpha)}\log\left(\frac{C_1}{(C_1-1)\alpha}\right)\right) = \frac{C_2C_1}{(C_1-\alpha)}\log\left(\frac{C_1}{(C_1-1)\alpha}\right)$. Then, from the initial claim we have that $n > \frac{C_2C_1}{(C_1-\alpha)}\log\left(\frac{C_1}{(C_1-1)\alpha}\right)$ implies $n > \frac{C_1}{(C_1-1)\alpha}\log(n)$. Also, $n > \frac{C_2C_1}{(C_1-\alpha)}\log\left(\frac{C_1}{(C_1-1)\alpha}\right)$ implies $n > C_1T$, which will imply $n - \frac{1}{\alpha}\log(n) > \frac{n}{C_1} > T$. Setting $\widetilde{\alpha}$ equals to $\frac{C_1}{(C_1-1)\alpha}$ proves the result. □

## D Identifiability Result

*Proof.* We first consider the case where there is only one non-trivial block $\mathcal{B}^{NT}$ inside $G$ and that the block cut vertices of $\mathcal{B}^{NT}$ do not have neighboring leaf nodes. As a result, $\mathcal{B}^{NT}$ contains exactly two block cut vertices $b_1$ and $b_2$ connected to the cut vertices $p_1$ and $p_2$, respectively. Thus, we express the vertex set $V$ of $G$ as a union of disjoint sets $V_1 \cup \{p_1\}$, $V_2 \cup \{p_2\}$, and $V_{NT}$—the vertex set of $\mathcal{B}^{NT}$.

Without loss of generality, let $V_1 \cup \{p_1\} = \{1, \ldots, p_1\}$, $V_{NT} = \{p_1+1, \ldots, p_2-1\}$, and $V_2 \cup \{p_2\} = \{p_2, \ldots, p\}$. Also, let $b_1 = p_1 + 1$ and $b_2 = p_2 - 1$. Because $G$, it follows that $V_1 \cup \{p_1\} \perp\!\!\!\perp V_1 \cup \{p_2\} \mid V_{NT}$. In words, $V_{NT}$ separates $V_1 \cup \{p_1\}$ and $V_2 \cup \{p_2\}$. Furthermore, $b_1$ shares an edge with $p_1$ and $b_2$ shares an edge with $p_2$. From these facts, $K^* = (\Sigma^*)^{-1}$ can be partitioned as in equation 17 (see below). Let $K_1$, $K_{NT}$, and $K_2$ be the first, second, and third diagonal blocks of $K^*$ in equation 17. Let $e_j$ be the canonical basis vector in $\mathbb{R}^p$. Then, we can express $K^*$ in equation 17 as

$$K^* = \text{Blkdiag}(K_1, K_{NT}, K_2) + e_{p_1+1}e_{p_1}^\intercal K_{p_1+1,p_1} + e_{p_1}e_{p_1+1}^\intercal K_{p_1,p_1+1} + e_{p_2-1}e_{p_2}^\intercal K_{p_2-1,p_2} + e_{p_2}e_{p_2-1}^\intercal K_{p_2,p_2-1}. \quad (18)$$

Recall that $\Sigma^0 = \Sigma^* + D$. Decompose the diagonal matrix $D$ as $D = D^{(1)} + D^{(2)}$, where

$$D^{(1)} = \text{Blkdiag}(\mathbf{0}, D^{(1)}_{NT}, \mathbf{0}), \quad (19)$$

$$D^{(2)} = \text{Blkdiag}(D_1, D^{(2)}_{NT}, D_2), \quad (20)$$

$$K^* = \left[\begin{array}{ccc|ccc|ccc}
K_{11} & \cdots & K_{1,p_1} & 0 & \cdots & 0 & & & \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & & \\
K_{p_1,1} & \cdots & K_{p_1,p_1} & K_{p_1+1,p_1} & \cdots & 0 & & & \\
\hline
0 & \cdots & K_{p_1,p_1+1} & K_{p_1+1,p_1+1} & \cdots & K_{p_1+1,p_2-1} & 0 & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & K_{p_2-1,p_1+1} & \cdots & K_{p_2-1,p_2-1} & K_{p_2-1,p_2} & \cdots & 0 \\
\hline
 & & & 0 & \cdots & K_{p_2,p_2-1} & K_{p_2,p_2} & \cdots & K_{p_2,p} \\
 & & & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
 & & & 0 & \cdots & 0 & K_{p,p_2} & \cdots & K_{p,p}
\end{array}\right] \quad (17)$$

and the dimensions of $D_1$, $D_{NT}$, and $D_2$ are same as those of $K_1$, $K_{NT}$, and $K_2$, resp. Furthermore, $D^{(1)}_{NT} = \text{diag}(0, \times, \ldots, \times, 0)$ and $D^{(2)}_{NT} = \text{diag}(\times, 0, \ldots, 0, \times)$. Here $\times$ can be a zero or a positive value. Let $\Sigma^q = \Sigma^* + D^{(1)}$ and $D^q = D^{(2)}$. From the above notations, we have $\Sigma^0 = \Sigma^* + D = \Sigma^* + D^{(1)} + D^{(2)} = \Sigma^q + D^q$. We show that there exists a decomposition of $D$ into $D_1$ and $D_2$ such that the inverse of $\Sigma^q \triangleq \Sigma^* + D^{(1)}$ has different structure. It suffices to show that $(\Sigma^q)^{-1}$ exactly equals the expression of $K^*$ in equation 18, except for the second diagonal block $K_{NT}$ in $\text{Blkdiag}(K_1, K_{NT}, K_2)$. Recall that different values of $K_{NT}$ yield different subgraphs on the non-trivial block, and consequently, different graphs in $[G]$; see Definition 3.1. Consider the following identity:

$$(\Sigma^q)^{-1} = (\Sigma^* + D^{(1)})^{-1} = (I + (\Sigma^*)^{-1} D^{(1)})^{-1} (\Sigma^*)^{-1} = (I + K^* D^{(1)})^{-1} K^*. \tag{21}$$

We first evaluate $(I + K^* D^{(1)})^{-1}$. Note that $e_{p_1+1}$, $e_{p_1}$, $e_{p_2-1}$, and $e_{p_2}$ lie in the nullspace of $D^{(1)}$ and $K^* D^{(1)}$. Using this fact and the formulas in equation 18 and equation 19, we can simplify $(I + K^* D^{(1)})$ as

$$(I + K^* D^{(1)}) = \text{Blkdiag}(I, I + K_{NT} D^{(1)}_{NT}, I), \tag{22}$$

where, $\widetilde{K}_{NT} \triangleq I + K_{NT} D^{(1)}_{NT}$ is a positive definite matrix, and hence, invertible. This is because $K_{NT} D^{(1)}_{NT}$ and $(D^{(1)}_{NT})^{1/2} K^{1/2}_{NT} K^{1/2}_{NT} (D^{(1)}_{NT})^{1/2}$ are similar matrices, where we used the facts that $K_{NT}$ is positive definite and $D^{(1)}_{NT}$ is non-negative diagonal. Thus,

$$(I + K^* D^{(1)})^{-1} = \text{Blkdiag}(I_{p_1}, \widetilde{K}^{-1}_{NT}, I_{p-p_2+1}). \tag{23}$$

Also, note that the null space vectors $e_{p_1+1}$, $e_{p_1}$, $e_{p_2-1}$, and $e_{p_2}$ of $K^* D^{(1)}$ are also the eigenvectors of $(I + K^* D^{(1)})^{-1}$, with eigenvalues all being equal to one. Putting together the pieces, from equation 18, equation 21, and equation 23 we have $(\Sigma^q)^{-1} = (I + K^* D^{(1)})^{-1} K^*$ which equals to the following:

$$= \text{Blkdiag}(K_1, \widetilde{K}^{-1}_{NT} K_{NT}, K_2) + e_{p_1+1} e^{\mathsf{T}}_{p_1} K^q_{p_1+1,p_1} + e_{p_1} e^{\mathsf{T}}_{p_1+1} K^q_{p_1,p_1+1} + e_{p_2-1} e^{\mathsf{T}}_{p_2} K^q_{p_2-1,p_2} + e_{p_2} e^{\mathsf{T}}_{p2-1} K^q_{p_2,p_2-1}.$$

Moreover, $\widetilde{K}^{-1}_{NT} K_{NT} = (I + K_{NT} D^{(1)}_{NT})^{-1} K_{NT} = (\Sigma_{NT} + D^{(1)}_{NT})^{-1}$, where $\Sigma_{NT} = K^{-1}_{NT}$ is the covariance of the random vector associated with $\mathcal{B}^{NT}$. Thus, $K^*$ in equation 18 and $(\Sigma^q)^{-1}$ are identical, except in their second diagonal blocks, as required. Furthermore, in order for the subgraph associated with $\widetilde{K}_{NT}$ to be a tree the entries in $\Sigma^q$ needs to be such that it matches the correlation factorization propert of a tree-subgraph. Using similar arguments, we can handle multiple internal blocks with block cut vertices that are not adjacent to leaf nodes. In the case where blocks have leaf nodes, we can combine the construction above with the construction in (Katiyar et al., 2019, Theorem 1) for tree structured graphical models. Combining these two, we can show that we can choose a decomposition $D = D_1 + D_2$ such that (a) the structure is arbitrarily different inside blocks, and (b) the block cut vertices are preserved (i.e., same as the ones in $G$), except they may be swapped with a neighboring leaf. □