Identification of Nonlinear Latent Hierarchical Models

Lingjing Kong[1], **Biwei Huang**[2], **Feng Xie**[3], **Eric Xing**[1,4], **Yuejie Chi**[1], and **Kun Zhang**[1,4]

[1] Carnegie Mellon University
[2]University of California San Diego
[3]Beijing Technology and Business University
[4]Mohamed bin Zayed University of Artificial Intelligence

Abstract

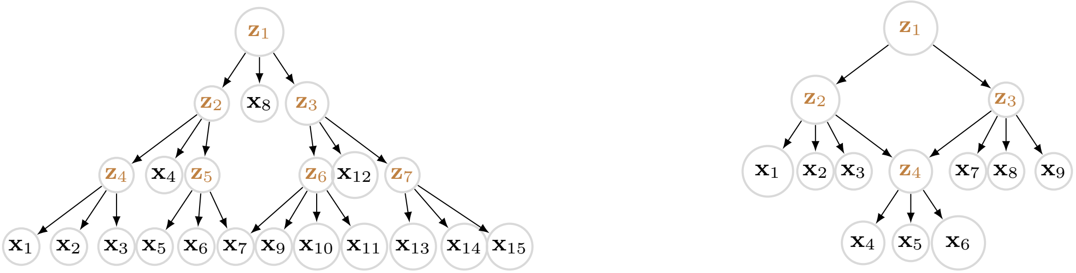
Identifying latent variables and causal structures from observational data is essential to many real-world applications involving biological data, medical data, and unstructured data such as images and languages. However, this task can be highly challenging, especially when observed variables are generated by causally related latent variables and the relationships are nonlinear. In this work, we investigate the identification problem for nonlinear latent hierarchical causal models in which observed variables are generated by a set of causally related latent variables, and some latent variables may not have observed children. We show that the identifiability of both causal structure and latent variables can be achieved under mild assumptions: on causal structures, we allow for the existence of multiple paths between any pair of variables in the graph, which relaxes latent tree assumptions in prior work; on structural functions, we do not make parametric assumptions, thus permitting general nonlinearity and multi-dimensional continuous variables. Specifically, we first develop a basic identification criterion in the form of novel identifiability guarantees for an elementary latent variable model. Leveraging this criterion, we show that *both causal structures and latent variables* of the hierarchical model can be identified asymptotically by explicitly constructing an estimation procedure. To the best of our knowledge, our work is the first to establish identifiability guarantees for both causal structures and latent variables in nonlinear latent hierarchical models.

1 Introduction

Classical causal structure learning algorithms often assume no latent confounders. However, in real-world scenarios, it is usually impossible to enumerate and measure all task-related variables. Neglecting latent confounders may lead to spurious correlations among observed variables. Hence, much effort has been devoted to handling the confounding problem. For instance, FCI and its variants [30, 26, 10, 2] leverage conditional independence constraints to locate possible latent confounders and estimate causal structures among observed variables, assuming no causal relationships among latent variables. This line of approaches ends up with an equivalence class, which usually consists of many directed acyclic graphs (DAGs).

Later, several types of approaches have been proposed to tackle direct causal relations among latent variables. These approaches are built upon principles such as rank constraints [29, 22, 15], matrix decomposition [6, 7, 3], high-order moments [28, 4, 34, 1, 8], copula models [11], and mixture oracles [20]. Pearl [25], Zhang [38], Choi et al. [9], Drton et al. [13], Zhou et al. [40], Huang et al. [16] extend such approaches to handle tree structures where only one path is permitted between any pair of variables. Recently, Huang et al. [15] propose to use rank-deficiency constraints to identify more general latent hierarchical structures. A common assumption behind these approaches is that either the causal relationships should be linear or the variables should be discrete. However, the linearity assumption and the discrete-variable assumption are rather restrictive in real-world scenarios.

Preprint. Under review.



(a) (b)

Figure 1: **Examples of latent hierarchical graphs satisfying Conditions 2.3**. $\mathbf{x}_i$ denotes observed variables and $\mathbf{z}_j$ denotes latent variables. The node size represents the dimensionality of each variable. We note that our graph condition permits multiple paths between two latent variables (e.g., $\mathbf{z}_1$ and $\mathbf{z}_4$ in Figure 1b), thus more general than tree structures.

Moreover, these approaches only focus on structure learning without identifying the latent variables and corresponding causal models. For a detailed literature review, please refer to Appendix A.

In this work, we identify the hierarchical graph structure and latent variables simultaneously for general nonlinear latent hierarchical causal models. Specifically, we first develop novel identifiability guarantees on a specific latent variable model which we refer to as the basis model (Figure 2). We then draw connections between the basis-model identifiability and the identification of nonlinear latent hierarchical models. Leveraging this connection, we develop an algorithm to localize and estimate latent variables and simultaneously learn causal structures underlying the entire system. We show the correctness of the proposed algorithm and thus obtain identifiability of both the *latent hierarchical causal structure* and *latent variable variables* for nonlinear latent hierarchical models. In sum, our main contributions are as follows.

- Analogous to independence tests in PC algorithm [30] and rank-deficiency tests in Huang et al. [15], we develop a novel identifiability theory (Theorem 3.2) as a fundamental criterion for locating and identifying latent variables in general nonlinear causal models.
- We show structure identification guarantees for latent hierarchical models admitting continuous multi-dimensional variables (c.f., one-dimensional and often discrete [25, 9, 15, 35]), general nonlinear (c.f., linear [25, 15, 35]) structural functions, and general graph structures (c.f., generalized trees [25, 9, 15]).
- Under the same conditions, we provide identification guarantees for latent variables, which to the best of our knowledge is the first attempt in cases of nonlinear hierarchical models.
- We accompany our theory with an estimation method that can asymptotically identify the causal structure and latent variables for nonlinear latent hierarchical models and validate it on multiple synthetic and real-world datasets.

2 Nonlinear Latent Hierarchical Causal Models

In this section, we introduce notations and formally define the latent hierarchical causal model. We focus on causal models with directed acyclic graph (DAG) structures and denote the graph with $\mathbf{G}$, which consists of the latent variable set $\mathbf{Z} := \{\mathbf{z}_1, \dots, \mathbf{z}_m\}$, the observed variable set [1] $\mathbf{X} := \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, and the edge set $\mathbf{E}$. Each variable is a random vector comprising potentially multiple dimensions, i.e., $\mathbf{x}_i \in \mathbb{R}^{d_{x_i}}$ and $\mathbf{z}_j \in \mathbb{R}^{d_{z_j}}$, where d_{x_i} and d_{z_j} stand for the dimensionalities of $\mathbf{x}_i \in \mathbf{X}$ and $\mathbf{z}_j \in \mathbf{Z}$, respectively. Both latent variables and observed variables are generated recursively by their latent parents:

$$\mathbf{x}_i = g_{x_i}(\text{Pa}(\mathbf{x}_i), \varepsilon_{x_i}) \qquad \mathbf{z}_j = g_{z_j}(\text{Pa}(\mathbf{z}_j), \varepsilon_{z_j}), \tag{1}$$

where $\text{Pa}(\mathbf{z}_j)$ represents all the parent variables of $\mathbf{z}_j$ and ε_{z_j} represents the exogenous variable generating $\mathbf{z}_j$. Identical definitions apply to those notations involving $\mathbf{x}_i$. Exogenous variables ε_{x_i}, ε_{z_j} are independent of each other and could also comprise multiple dimensions. We now define a general latent hierarchical causal model with a causal graph $\mathbf{G} := (\mathbf{Y}, \mathbf{E})$ in Definition 2.1.

Definition 2.1 (Latent hierarchical causal model)**.**

i The variable set $\mathbf{Y}$ consists of the observed variable set $\mathbf{X}$ and the latent variable set $\mathbf{Z}$, i.e., $\mathbf{Y} := \mathbf{X} \cup \mathbf{Z}$.

[1] We refer to leaf variables in the graph as observed variables for ease of exposition. The theory in this work is also applicable when some non-leaf variables happen to be observed, which we discuss in Append C.2

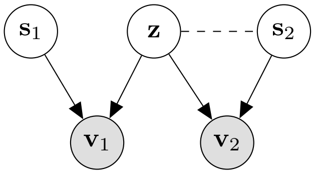


Figure 2: **The basis model**. $\mathbf{v}_1$ and $\mathbf{v}_2$ are generated from nonlinear functions g_1 and g_2 which can be distinct and non-invertible individually. We also admit dependence between $\mathbf{z}$ and $\mathbf{s}_2$, as indicated by the dashed edge.

ii Each variable in $\mathbf{Y}$ follows the generating process described in Equation 1.

iii The distribution over $\mathbf{Y}$ is Markovian and faithful to the graph $\mathbf{G}$.

In light of the given definitions, we formally state the objectives of this work:

1. Structure identification: given observed variables $\mathbf{X}$, we would like to recover the edge set $\mathbf{E}$.
2. Latent-variable identification: given observed variables $\mathbf{X}$, we would like to obtain a set of variables $\hat{\mathbf{Z}} := \{\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_m\}$ where $\mathbf{z}_i$ and $\hat{\mathbf{z}}_i$ are identical up to an invertible mapping for $i \in [m]$, i.e., $\hat{\mathbf{z}}_i = h_i(\mathbf{z}_i)$, where h_i is an invertible function.

Definition 2.1 gives a general class of latent hierarchical causal models. On the functional constraint, the general nonlinear function form (Equation 1) renders it highly challenging to identify either the graph structure or the latent variables. Prior work [25, 9, 35, 15] relies on either linear model conditions or discrete variable assumptions. In this work, we remove these conditions to address the general nonlinear case with continuous variables. On the structure constraint, generally speaking, the identification of arbitrary causal structures is impossible. For instance, tree-like structures are assumed in prior work [25, 9, 15] to obtain structural identifiability. In the following, we present more general structural conditions for which we show identifiability.

Definition 2.2 (Pure children)**.** $\mathbf{v}_i$ is a pure child of another variable $\mathbf{v}_j$, if $\mathbf{v}_j$ is the only parent of $\mathbf{v}_i$ in the graph, i.e., Pa$(\mathbf{v}_i) = \{\mathbf{v}_j\}$.

Condition 2.3 (Structural conditions)**.**

i Each latent variable has at least 2 pure children.

ii There are no directed paths between any two siblings.

Intuitively, Condition 2.3-i allows each latent variable to leave a sufficiently large footprint for identification. This excludes some fundamentally unidentifiable structures. For instance, a latent variable with no pure children can be merged into its children without affecting the overall generating process. We require fewer pure children than prior work [29, 22] and place no constraint on the number of neighboring variables as in [15, 35].

Condition 2.3-ii avoids potential triangle structures in the latent hierarchical model. Triangles present significant obstacles for identification – in a triangle structure formed by $\mathbf{z}_1 \to \mathbf{z}_2 \to \mathbf{z}_3$ and $\mathbf{z}_1 \to \mathbf{z}_3$, it is difficult to discern how $\mathbf{z}_1$ influences $\mathbf{z}_3$ without functional constraints, as there are two directed paths between them. We defer the discussion of how to use our techniques to handle more general graphs that include triangles to Appendix C.1. Condition 2.3-ii is widely adopted in prior work, especially those on tree structures [25, 9, 13] (which cannot handle multiple undirected paths between variables as we do) and more recent work [15]. To the best of our knowledge, only Xie et al. [35] manage to handle triangles in the latent hierarchical structure, which, however, heavily relies on strong assumptions on both the function class (linear functions), distribution (non-Gaussian), and structures (the existence of neighboring variables).

3 Identifiability of Nonlinear Latent Hierarchical Causal Models

This section presents the identifiability and identification procedure of causal structures and latent variables in nonlinear latent hierarchical causal models, from only observed variables $\mathbf{X}$. First, we introduce a latent variable model (i.e., the *basis model*) as illustrated in Figure 2, whose identifiability underpins the identifiability of the hierarchical model. By establishing the connection between the basis model and the hierarchical model in Theorem 3.4, we show by construction that both the causal structure and latent variables are identifiable in the hierarchical model in Theorem 3.5.

3.1 Basis Model Identifiability

We present the data-generating process of the basis model in Equation 2 and illustrate it in Figure 2:

$$\mathbf{v}_1 = g_1(\mathbf{z}, \mathbf{s}_1) \qquad \mathbf{v}_2 = g_2(\mathbf{z}, \mathbf{s}_2), \tag{2}$$

where $\mathbf{v}_1 \in \mathcal{V}_1 \subset \mathbb{R}^{d_{v_1}}$, $\mathbf{v}_2 \in \mathcal{V}_2 \subset \mathbb{R}^{d_{v_2}}$, $\mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^{d_z}$, $\mathbf{s}_1 \in \mathcal{S}_1 \subset \mathbb{R}^{d_{s_1}}$, and $\mathbf{s}_2 \in \mathcal{S}_2 \subset \mathbb{R}^{d_{s_2}}$ are variables consisting of potentially multiple components. The data-generating process admits potential dependence between $\mathbf{z}$ and $\mathbf{s}_2$ and dependence across dimensions in each variable. We denote the entire latent variable $\mathbf{c} := [\mathbf{z}, \mathbf{s}_1, \mathbf{s}_2] \in \mathcal{C}$ and the mapping from $\mathbf{c}$ to $(\mathbf{v}_1, \mathbf{v}_2)$ as $g : \mathcal{C} \to \mathcal{V}_1 \times \mathcal{V}_2$.

Below, we first introduce notations and basic definitions that we use throughout this work.
Indexing: For a matrix $\mathbf{M}$, we denote its i-th row as $\mathbf{M}_{i,:}$, its j-th column as $\mathbf{M}_{:,j}$, and its (i, j) entry as $\mathbf{M}_{i,j}$. Similarly, for an index set $\mathcal{I} \subseteq \{1, \dots, m\} \times \{1, \dots, n\}$, we denote $\mathcal{I}_{i,:} := \{j | (i, j) \in \mathcal{I}\}$ and $\mathcal{I}_{:,j} := \{i | (i, j) \in \mathcal{I}\}$.
Subspaces: We denote a subspace of $\mathbb{R}^n$ defined by an index set $\mathcal{S}$ as $\mathbb{R}^n_{\mathcal{S}}$, where $\mathbb{R}^n_{\mathcal{S}} := \{\mathbf{z} \in \mathbb{R}^n : \forall i \notin \mathcal{S}, z_i = 0\}$.
Support of a matrix-valued function: We define the support of a matrix-valued function $\mathbf{M}(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^{m \times n}$ as Supp$(\mathbf{M}) := \{(i, j) : \exists \mathbf{x} \in \mathcal{X}, \text{s.t.}, \mathbf{M}(\mathbf{x})_{i,j} \neq 0\}$, i.e., the set of indices whose corresponding entries are nonzero for some $\mathbf{x} \in \mathcal{X}$.

In Theorem 3.2 below, we show that the latent variable $\mathbf{z}$ shared by $\mathbf{v}_1$ and $\mathbf{v}_2$ is identifiable up to an one-to-one mapping by estimating a generative model $(\hat{p}_{\mathbf{z},\mathbf{s}_2}, \hat{p}_{\mathbf{s}_1}, \hat{g})$ according to Equation 2. We denote the support of Jacobian matrix $\mathbf{J}_g$ and $\mathbf{J}_{\hat{g}}$ as $\mathcal{G}$ and $\hat{\mathcal{G}}$ respectively and denote by $\mathbf{T}$ a matrix with the same support as $\mathbf{T}(\mathbf{c})$ in $\mathbf{J}_{\hat{g}}(\hat{\mathbf{c}}) = \mathbf{J}_g(\mathbf{c})\mathbf{T}(\mathbf{c})$.

Assumption 3.1 (Basis model conditions)**.**

i *[Invertibility]: The joint mapping $g(\mathbf{c}) = (\mathbf{v}_1, \mathbf{v}_2)$ is invertible.*

ii *[Subspace span]: For all $i \in \{1, \dots, d_{v_1} + d_{v_2}\}$, there exists $\{\mathbf{c}^{(\ell)}\}_{\ell=1}^{|\mathcal{G}_{i,:}|}$, such that span$(\{\mathbf{J}_g(\mathbf{c}^{(\ell)})_{i,:}\}_{\ell=1}^{|\mathcal{G}_{i,:}|}) = \mathbb{R}^{d_c}_{\mathcal{G}_{i,:}}$ and $[\mathbf{J}_g(\mathbf{c}^{(\ell)})\mathbf{T}]_{i,:} \in \mathbb{R}^{d_c}_{\hat{\mathcal{G}}_{i,:}}$.*

iii *[Edge Connectivity]: For all $j_z \in \{1, \dots, d_z\}$, there exist i_{v_1} and i_{v_2}, such that $(i_{v_1}, j_z) \in \mathcal{G}$ and $(i_{v_2}, j_z) \in \mathcal{G}$.*

Theorem 3.2. *Under Assumption 3.1, if a generative model $(\hat{p}_{\mathbf{z},\mathbf{s}_2}, \hat{p}_{\mathbf{s}_1}, \hat{g})$ follows the data-generating process in Equation 2 and matches the true joint distribution:*

$$p_{\mathbf{v}_1,\mathbf{v}_2}(\mathbf{v}_1, \mathbf{v}_2) = \hat{p}_{\mathbf{v}_1,\mathbf{v}_2}(\mathbf{v}_1, \mathbf{v}_2), \forall (\mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V} \times \mathcal{V}, \tag{3}$$

then the estimated variable $\hat{\mathbf{z}}$ and the true variable $\mathbf{z}$ are equivalent up to a one-to-one mapping.

The proof can be found in Appendix B.1.

Interpretation. Intuitively, we can infer the shared latent variable $\mathbf{z}$ from only observational data $(\mathbf{v}_1, \mathbf{v}_2)$, in a sense that the estimated variable $\hat{\mathbf{z}}$ contains all the information of $\mathbf{z}$ without mixing any information from $\mathbf{s}_1$ and $\mathbf{s}_2$. This notion of blockwise-identifiability is extensively employed in identifiability theory [5, 17, 23, 32, 33, 21].

Discussion on assumptions. Assumption 3.1-i assumes the overall invertibility and is strictly weaker than the invertibility of each causal model as in [33, 24].

Assumption 3.1-ii guarantees that the influence of $\mathbf{z}$ changes adequately across its domain. This eliminates situations where the Jacobian matrix is partially constant, causing it to insufficiently capture the connection between the observed and latent variables. To circumvent such situations and make the assumption hold, one can restructure latent variables by merging redundant components.

Assumption 3.1-iii is a formal characterization of the common cause variable z in the basis model. This assumption excludes the scenario where some dimensions of $\mathbf{z}$ only influence one of $\mathbf{v}_1$ and $\mathbf{v}_2$, in which case these dimensions of $\mathbf{z}$ are not truly the common cause of $\mathbf{v}_1$ and $\mathbf{v}_2$ and should be modeled as a separate variable rather than part of $\mathbf{z}$.

Comparison to prior work. Theorem 3.2 allows the most general conditions, comparing to similar data-generating processes in prior work [24, 33]. First, Theorem 3.2 does not necessitate the invertibility for $g_1(\cdot)$ and $g_2(\cdot)$ individually as in [24, 33]. Essentially, the invertibility of g_1 and g_2 amounts to asking that the information of $\mathbf{z}$ is duplicated and stored in both $\mathbf{v}_1$ and $\mathbf{v}_2$. In contrast, Assumption 3.1-i only requires that information is stored in $(\mathbf{v}_1, \mathbf{v}_2)$ jointly. In contrast to the identical generating function assumption $g_1(\cdot) = g_2(\cdot)$ in [33], the two generating functions $g_1(\cdot)$ and $g_2(\cdot)$ can be distinct in Theorem 3.2. Additionally, Theorem 3.2 allows for dependence between $\mathbf{z}$ and $\mathbf{s}_2$ which is absent in [24].

These relaxations significantly expand the applicability of the basis model. For example, the distinction between g_1 and g_2 is indispensable in our application to the hierarchical model, as we cannot assume each latent variable generates its many children and descents through the same function. This is also the case for many other causal structure/variable learning problems, such as measurement models, which we discuss in Section 3.2. Moreover, our proof techniques are distinct from prior work along this line, which can be of independent interest to the community.

3.2 Local Identifiability of Latent Hierarchical Models

In this section, we build the connection between the basis model and the hierarchical model. Specifically, we show in Theorem 3.4 that a careful grouping of variables in the hierarchical model enables us to apply Theorem 3.2 to identify latent variables and their causal relations in a local scope.

To invoke Theorem 3.2, we adjust Assumption 3.1 in basis models to obtain Assumption 3.3 for hierarchical models.

Assumption 3.3 (Hierarchical model conditions)**.**

i *[Information-conservation]: Any $\mathbf{z} \in \mathbf{Z}$ and exogenous variable $\boldsymbol{\varepsilon}$ can be expressed as functions of all observed variables, i.e., $\mathbf{z} = f_z(\mathbf{X})$ and $\boldsymbol{\varepsilon} = f_\varepsilon(\mathbf{X})$.*

ii *[Subspace span]: For each variable $\mathbf{z}$ and any set $\mathbf{A_z} \subset \mathbf{Y}$ such that 1) $\mathbf{z} \perp\!\!\!\perp \mathbf{X} | \mathbf{A_z}$ and 2) $\mathbf{A_z}$ does not contain directed paths, any invertible function $\tilde{g} : \mathbb{R}^m \to \mathbb{R}^n$ with $\mathbf{z}$ included in its input and $\mathbf{A_z}$ as its output satisfies the subspace span condition (i.e., Assumption 3.1-ii).*

iii *[Edge connectivity]: The function between each latent variable $\mathbf{z}$ and each of its children $\mathbf{z}'$ has a Jacobian $\mathbf{J}$, such that for all $j \in \{1, \ldots, d_z\}$, there exists $i \in \{1, \ldots, d_{z'}\}$, such that (i, j) is in the support of $\mathbf{J}$.*

Theorem 3.4. *In a latent hierarchical causal model that satisfies Condition 2.3 and Assumption 3.3, we consider $\mathbf{x}_i \in \mathbf{X}$ as $\mathbf{v}_1$ and $\mathbf{X} \backslash \mathbf{v}_1$ as $\mathbf{v}_2$ in the basis model (Figure 2).* [2] *With an estimation model $(\hat{p}_{\mathbf{z},\mathbf{s}_2}, \hat{p}_{\mathbf{s}_1}, \hat{g})$ that follows the data-generating process in Equation 2, the estimated $\hat{\mathbf{z}}$ is a one-to-one mapping of the parent(s) of $\mathbf{v}_1$, i.e., $\hat{\mathbf{z}} = h(Pa(\mathbf{v}_1))$ where $h(\cdot)$ is an invertible function.*
The proof can be found in Appendix B.2.

Interpretation. Theorem 3.4 shows that invoking Theorem 3.2 with the assignment $\mathbf{v}_1 = \mathbf{x}$ and $\mathbf{v}_2 = \mathbf{X} \backslash \{\mathbf{x}\}$ can identify the parents of $\mathbf{x}$ in the hierarchical model. Intuitively, we can identify latent variables one "level" above the current level $\mathbf{X}$. Moreover, as the estimated variables are equivalent to true variables up to one-to-one mappings, we take a step further in Section 3.3 to show that this procedure can be applied to the newly estimated variables iteratively to traverse the hierarchical model "level" by "level". In the degenerate case where $\mathbf{v}_1$ does not have any parents, i.e., the root of the hierarchical model, the identified variable $\mathbf{z}$ in the basis model corresponds to the $\mathbf{v}_1$ itself – we can regard $\mathbf{s}_1$ as a constant and $g_1(\cdot)$ as an identity function in Theorem 3.2.

Discussion on assumptions. Assumption 3.3-i, corresponding to Assumption 3.1-i for the basis model, is a minimal form of invertibility for the identifiability of latent variables. Intuitively, it states that the information in the system is preserved without unnecessary duplication. Without this, the information of latent variables would be lost through the transformations and cannot be identified.

Assumption 3.3-ii is a direct extension of Assumption 3.1-ii to the hierarchical case, with $\mathbf{A_z}$ playing the role of $(\mathbf{v}_1, \mathbf{v}_2)$. Thus, it is a general condition as discussed for Assumption 3.1-ii.

Assumption iii excludes degenerate cases where an edge connects components that do not influence each other. This can always be fulfilled by explicitly modeling such inactive connectivity in the graph. Further, when $\mathbf{v}_1 = \mathbf{x}$ is a pure child of a latent variable $\mathbf{z}$, Condition 2.3-i ensures that $\mathbf{v}_2$ always contains at least one pure child or descendant of $\mathbf{z}$ to fulfill Assumption 3.1-iii.

Implications on measurement causal models. Despite being an intermediate step towards the global structure identification, Theorem 3.4 is a standalone contribution: it can be applied to general nonlinear measurement models with arbitrary latent structures and identify all the latent variables. Then, the latent structures in the measurement model can be identified by performing existing causal discovery algorithms, such as the PC algorithm [31], on the identified latent variables. We leave the detailed presentation of this application as future work.

3.3 Global Identifiability of Latent Hierarchical Causal Models

Equipped with the local identifiability (i.e., Theorem 3.4), the following theorem shows that *all* causal structures and latent variables are identifiable in the hierarchical model.

[2] To avoid cluttering, we slightly abuse the bold lowercase font to represent either an individual vector or a vector set (which can be viewed as a concatenation).

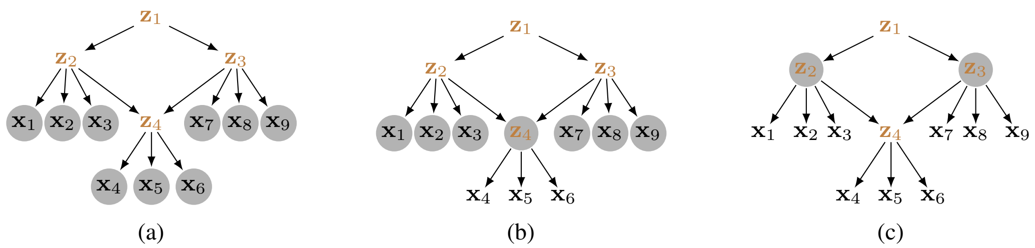


Figure 3: **Evolution of active set $\mathbf{A}$ in Algorithm 1.** We mark the active set $\mathbf{A}$ with shaded gray circles before each iterations of Algorithm 1, with Figure 3a, Figure 3b, and Figure 3c corresponding to iteration 1, 2, and 3. Before iteration 1, $\mathbf{A}$ is set to $\mathbf{X}$ by default. At iteration 1, $\mathbf{z}_2$, $\mathbf{z}_3$, and $\mathbf{z}_4$ can be estimated by the basis model; however, only $\mathbf{z}_4$ can be updated into $\mathbf{A}$. Otherwise, directed paths would be introduced by $\mathbf{z}_2$ and $\mathbf{z}_3$.

Theorem 3.5. *Under assumptions in Theorem 3.4, all latent variables $\mathbf{Z}$ in the hierarchical model can be identified up to one-to-one mappings, and the causal structure $\mathbf{G}$ can also be identified.*

Comparison to prior work. Theorem 3.5 handles general nonlinear cases, whereas prior work [25, 9, 15, 35] is limited to linear function or discrete latent variable conditions. Structure-wise, our identifiability results apply to latent structures with V-structures and potentially triangle structures beyond generalized trees as studied in prior work [9, 25, 15], require fewer pure children and no neighboring variables in comparison with [35, 15], and can determine directions for each edge (c.f., Markov-equivalent classes in Huang et al. [15]).

Proof sketch with search procedures. The proof can be found in Appendix B.3. In particular, we develop a concrete algorithm (i.e., Algorithm 1) and show that it can successfully identify the causal structure and latent variables asymptotically. We give a proof sketch of Theorem 3.4 below by navigating through Algorithm 1 and illustrate it with an example in Figure 3.

Stage 1: identifying parents with the basis model (line 4-line 5). As shown in Theorem 3.4, the basis model can identify the latent parents of leaf-level variables in the hierarchy. In Figure 3a, we can identify $\mathbf{z}_2$ when applying the basis model with the assignment $\mathbf{v}_1 := \{\mathbf{x}_1\}$ and $\mathbf{v}_2 := \mathbf{X} \setminus \mathbf{v}_1$. Naturally, the basic idea of Algorithm 1 is to iteratively apply the basis model to the most recently identified latent variables to further identify their parents. In Algorithm 1, we define as active set $\mathbf{A}$ the set of variables to which we apply the basis model. For example, $\mathbf{A}$ equals to $\mathbf{X}$ in the first round (Figure 3a) and becomes $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{z}_4, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9\}$ in the second round (Figure 3b).

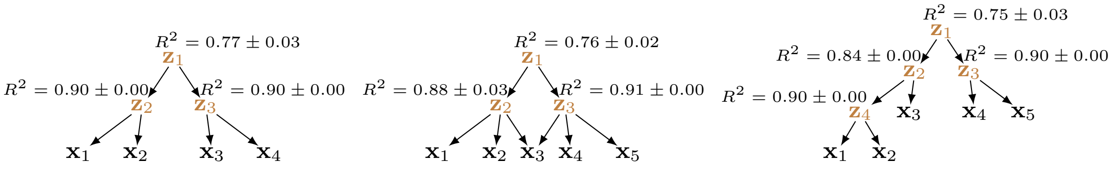
Stage 2: merging duplicate variables (line 6). As multiple variables in $\mathbf{A}$ can share a parent, dictionary $\mathbf{P}(\cdot)$ may contain multiple variables that are one-to-one mappings to each other. Line 6 detects such duplicates and represents them with one variable. In Figure 3a, setting $\mathbf{v}_1$ to any of $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ would yield an equivalent variable of $\mathbf{z}_2$. We merge the three equivalents by randomly selecting one.

Stage 3: detecting and merging super-variables (line 7 - line 9). Due to the potential existence of V-structures, variables in $\mathbf{A}$ may have multiple parents and produce super-variables in $\mathbf{P}$. For instance, at the second iteration of Algorithm 1 (i.e., Figure 3b), the basis model will be run on $\mathbf{v}_1 = \mathbf{z}_4$ and $\mathbf{v}_2 = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9\}$ and output an variable equivalent to the concatenation $(\mathbf{z}_2, \mathbf{z}_3)$. Line 7 to line 9 detect such super-variables by testing whether each variable $\hat{\mathbf{z}}$ in $\mathbf{P}$ is equivalent to a union of other variables in $\mathbf{P}$. If such a union $\tilde{\mathbf{Z}} := \{\hat{\mathbf{z}}_1, \dots, \hat{\mathbf{z}}_m\}$ exists, we will replace $\hat{\mathbf{z}}$ with $\tilde{\mathbf{Z}}$ in all its occurrences. In Figure 3b, we would split the variable equivalent to $[\mathbf{z}_2, \mathbf{z}_3]$ into variables of $\mathbf{z}_2$ and $\mathbf{z}_3$ individually. If $\hat{\mathbf{z}}$ is tested to be a super-variable, i.e., it can perfectly predict another variable $\hat{\mathbf{z}}'$ in $\mathbf{P}$ and $\hat{\mathbf{z}}'$ cannot predict $\hat{\mathbf{z}}$ perfectly, and the equivalent union cannot be found, we will track $\hat{\mathbf{z}}$ in line 9 to prevent it from being updated into $\mathbf{A}$ at line 16.

Stage 4: detecting and avoiding directed paths in $\mathbf{A}$ (line 12-line 14). Ideally, we would like to repeat line 4 to line 9 until reaching the root of the hierarchical model. Unfortunately, such an approach can be problematic, as this would cause variables in active set $\mathbf{A}$ to have directed paths among them, in contrast to leaf variables $\mathbf{X}$ in Theorem 3.4. In Figure 3a, $\mathbf{P}$ would contain $\{\mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4\}$, as each of these latent variables has pure observed children in $\mathbf{X}$. However, due to direct paths within $\mathbf{P}$, i.e., $\mathbf{z}_2 \to \mathbf{z}_4$ and $\mathbf{z}_3 \to \mathbf{z}_4$, we cannot directly substitute $\mathbf{X}$ with $\{\mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4\}$ in $\mathbf{A}$. To resolve this dilemma, we proactively detect directed paths emerging with newly estimated variables and defer the local update of such estimated variables to eliminate direct paths. For directed path detection, we introduce a corollary of Theorem 3.4 as Corollary 3.6.

Algorithm 1 Identification of Latent Hierarchical Models. $\mathbf{A}$: the active set, $\mathbf{X}$: the observed variable set, **P/JointP**: dictionaries that store the relations between variables in $\mathbf{A}$ and estimated variables.

```
1:  Initialize the active set: $\mathbf{A} \leftarrow \mathbf{X}$;
2:  while $\mathbf{A} \neq \emptyset$ do
3:      initialize an empty set $\mathbf{R}$ and empty dictionaries $\mathbf{P}$, JointP;
4:      for each active variable $\mathbf{a} \in \mathbf{A}$ do
5:          estimate the basis model with $\mathbf{v}_1 = \mathbf{a}$ and $\mathbf{v}_2 = \mathbf{A} \setminus \{\mathbf{a}\}$ to obtain $\hat{\mathbf{z}}$, and $\mathbf{P} \leftarrow \mathbf{P} \cup \{(\mathbf{a}, \hat{\mathbf{z}})\}$;
6:      merge equivalent variables in $\mathbf{P}$;
7:      for each variable $\hat{\mathbf{z}}$ in $\mathbf{P}$ do
8:          if $\exists \hat{\mathbf{z}}' \in \mathbf{P}$ that $\hat{\mathbf{z}}$ can perfectly predict but $\hat{\mathbf{z}}'$ cannot predict $\hat{\mathbf{z}}$ then
9:              if $\exists \tilde{\mathbf{Z}} \subseteq \mathbf{P}$ that perfectly predicts $\hat{\mathbf{z}}$, then replace $\hat{\mathbf{z}}$ with $\tilde{\mathbf{Z}}$; else: $\mathbf{R} \leftarrow \mathbf{R} \cup \{\hat{\mathbf{z}}\}$;
10:     cluster variables in $\mathbf{P}$ into $\{\mathbf{Z}_i\}_{i=1}^m$ where $\mathbf{Z}_i$ is either a singleton or contains spouse variables;
11:     store each cluster $\mathbf{Z}_i$ and its children set $\mathbf{H}_i$ as a pair $(\mathbf{Z}_i, \mathbf{H}_i)$ into JointP;
12:     for each pair $(\mathbf{Z}_i, \mathbf{H}_i) \in$ JointP do
13:         estimate the basis model with $\mathbf{v}_1 = \mathbf{Z}_i$ and $\mathbf{v}_2 = \mathbf{A} \setminus \mathbf{H}_i$ to obtain an variable $\hat{\mathbf{z}}_{\text{test}}$;
14:         if $\exists \mathbf{z}' \in \mathbf{Z}_i$ such that $\hat{\mathbf{z}}_{\text{test}}$ can perfectly predict $\mathbf{z}'$, then $\mathbf{R} \leftarrow \mathbf{R} \cup \mathbf{Z}_i$
15:     for each pair $(\mathbf{Z}_i, \mathbf{H}_i) \in$ JointP do
16:         if no variables in $\mathbf{Z}_i$ has been tracked by $\mathbf{R}$, i.e., $\mathbf{Z}_i \cap \mathbf{R} = \emptyset$, then substitute $\mathbf{H}_i$ with $\mathbf{Z}_i$ in $\mathbf{A}$.
17:     remove independent variables from $\mathbf{A}$.
18: Return: all the past active sets $\mathbf{A}$ and parent sets $\mathbf{P}$.
```



(a) A balanced tree.

(b) With V-structure.

(c) An unbalanced tree.

Figure 4: **Evaluated hierarchical models.** We denote the estimation R^2 scores around corresponding latent variables. We can observe that all latent variables can be identified decently, justifying our theoretic results.

Corollary 3.6. *Under assumptions in Theorem 3.4, for an arbitrary variable* $\mathbf{z}$ *from* $Pa(\mathbf{A})$*, we consider* $\mathbf{v}_1 := \mathbf{z}$ *and* $\mathbf{v}_2 := \mathbf{A} \setminus (Ch(\mathbf{z}) \cap \mathbf{A})$ *(i.e., the active set excluding* $\mathbf{z}$*'s identified children), estimating the basis model yields* $\hat{\mathbf{z}}$ *which is equivalent to* $\mathbf{z}$ *up to an one-to-one mapping.*

Corollary 3.6 can be obtained by setting $\mathbf{v}_1$ as an ancestor of $\mathbf{v}_2$ and $\mathbf{s}_1$ as a degenerate variable (i.e., a deterministic quantity) in Theorem 3.4. Leveraging Corollary 3.6, we can detect whether each newly identified latent variable in $\mathbf{P}$ would introduce directed paths into $\mathbf{A}$ if they were substituted in. If directed paths exist, the variable $\hat{\mathbf{z}}$ would contain the same information as $\mathbf{v}_1 := \mathbf{z}$, which prediction tests can evaluate. In this event, we will suppress the update of the $\mathbf{z}$ at this iteration. That is, we still keep its children in $\mathbf{A}$. This directed-path detection is conducted in lines 12- 14, after properly grouping variables that share children in lines 10- 11. As shown in Figure 3b, $\mathbf{z}_2$ and $\mathbf{z}_3$ are not placed in $\mathbf{A}$, even if they are found in the first iteration. This update only happens when $\mathbf{z}_4$ has been placed in $\mathbf{A}$ at the second iteration.

In general, a latent variable enters $\mathbf{A}$ only when all its children reside in $\mathbf{A}$. We can show that $\mathbf{A}$ contains all the information of unidentified latent variables – $\mathbf{A}$ d-separates the latent variables that have not been placed in $\mathbf{A}$ and those were in $\mathbf{A}$ once. Equipped with such a procedure, we can identify the hierarchical model iteratively until completion.

4 Experimental Results

In this section, we present experiments to corroborate our theoretical results in Section 3. We start with the problem of recovering the basis model in Section 4.2, which is the foundation of the overall identifiability. In Section 4.3, we present experiments for hierarchical models on a synthetic dataset and two real-world datasets.

4.1 Experimental Setup

Synthetic data generation. For the basis model (Figure 2), we sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{s}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\mathbf{s}_2 \sim \mathcal{N}(\mathbf{Az} + \mathbf{b}, \mathbf{I})$, where the dependence between $\mathbf{z}$ and $\mathbf{s}_2$ is implemented by randomly

	$d_z = d_{s_1} = d_{s_2} = 2$	$d_z = d_{s_1} = 2, d_{s_2} = 3$	$d_z = d_{s_1} = d_{s_2} = 4$	$d_z = d_{s_1} = 4, d_{s_2} = 6$
Joint invertibility (Ours)	0.94 ± 0.01	0.90 ± 0.05	0.89 ± 0.05	0.82 ± 0.14
Individual invertibility [33]	0.65 ± 0.01	NA	0.68 ± 0.01	NA

Table 1: **The basis model identifiability**. We show the identifiability for $\mathbf{z}$ under various data dimensionalities d_z, d_{s_1}, and d_{s_2} for $\mathbf{z}$, $\mathbf{s}_1$, $\mathbf{s}_2$. We compare our results with prior work that assumes both g_1 and g_2 are invertible individually. NA indicates that the model is not applicable when the dimensionalities of $\mathbf{s}_1$ and $\mathbf{s}_2$ differ.

constructed matrix $\mathbf{A}$ and bias $\mathbf{b}$. We choose the true mixing function $\mathbf{g}$ as a two-layer multilayer perceptron (MLP) with Leaky-ReLU activation functions. We apply element-wise $\max\{z, 0\}$ to $\mathbf{z}$ before inputting $[\mathbf{z}, \mathbf{s}_1]$ to g_1 and element-wise $\min\{z, 0\}$ to $\mathbf{z}$ before inputting $[\mathbf{z}, \mathbf{s}_2]$ to g_2, so that $\mathbf{v}_1$ is generated by the positive elements of $\mathbf{z}$ and $\mathbf{v}_2$ is generated by the negative elements of $\mathbf{z}$. This way, g_1 and g_2 are jointly invertible but not individually invertible. For latent hierarchical models (Figure 4), we sample each exogenous variable $\boldsymbol{\varepsilon}$ as $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and each endogenous variable $\mathbf{z}$ is endowed with a distinct generating function $g_{\mathbf{z}}$ parameterized by an MLP, i.e., $\mathbf{z} = g_{\mathbf{z}}(\text{Pa}(\mathbf{z}), \boldsymbol{\varepsilon}_{\mathbf{z}})$.

Real-world datasets. We adopt two real-world datasets with hierarchical generating processes, namely a personality dataset and a digit dataset. The personality dataset was collected through an interactive online personality test [27]. Participants were requested to provide a rating for each question on a five-point scale. Each question was designed to be associated with one of the five personality attributes, i.e., agreeableness, openness, conscientiousness, extraversion, and neuroticism. The corresponding answer scores are denoted as a_i, o_j, etc, as indicated in Figure 5. We use responses (around 19,500 for each question) to six questions for each of the five personality attributes. For the digit dataset, we construct a multi-view digit dataset from MNIST [12]. We first randomly crop each image to obtain two intermediate views and then randomly rotate each of the intermediate views independently to obtain four views. This procedure gives rise to a latent structure similar to that in Figure 4a. We feed images to a pretrained ResNet-44 [14] for dimensionality reduction ($28 \times 28 \rightarrow 64$) and run our algorithm on the produced features.

Estimation models. We implement the estimation model $(\hat{g}_1, \hat{g}_2, \hat{f})$ following Equation 2, where $\hat{f}$ can be seen as an encoder that transforms $(\mathbf{v}_1, \mathbf{v}_2)$ to the latent space and $(\hat{g}_1, \hat{g}_2)$ act as the decoders generating $\mathbf{v}_1$ and $\mathbf{v}_2$ respectively. We parameterize each module with an MLP with Leaky-ReLU activation. Training configurations can be found in Appendix D.

Evaluation metrics. Due to the block-wise nature of our identifiability results, we adopt the coefficient of determination R^2 between the estimated variables $\hat{\mathbf{z}}$ and the true variables $\mathbf{z}$, where $R^2 = 1$ suggests that the estimated variable $\hat{\mathbf{z}}$ can perfectly capture the variation of the true variable $\mathbf{z}$. We apply kernel regression with Gaussian kernel to estimate the nonlinear mapping. Such a protocol is employed in related work [33, 21]. We repeat each experiment over at least 3 random seeds.

4.2 Basis Model Identification

The results for the basis model are presented in Table 1 and Figure 9. We vary the number of components of each latent partition $(d_{\mathbf{z}}, d_{\mathbf{s}_1}, d_{\mathbf{s}_2})$. We can observe that the model with individual invertibility (as assumed in prior work [33, 24]) can only capture around half of the information in $\mathbf{z}$, due to their assumption that both g_1 and g_2 are invertible, which is violated in this setup. In contrast, our estimation approach can largely recover the information of $\mathbf{z}$ across a range of latent component dimensions, verifying our Theorem 3.2. Moreover, prior work [33] assumes $g_1 = g_2$, and therefore cannot be applicable when the dimensionalities of $\mathbf{s}_1$ and $\mathbf{s}_2$ differ (e.g., $d_{s_1} = 2$, $d_{s_2} = 3$), hence the “NA” in the table. Figure 9 in Appendix D.2 shows scatter-plots of the true and the estimated components with $d_z = d_{s_1} = d_{s_2} = 2$. We can observe that components of $\mathbf{z}$ and those of $\hat{\mathbf{z}}$ are highly correlated, suggesting that the information of $\mathbf{z}$ is indeed restored. In contrast, $\hat{\mathbf{z}}$ contains very little information of $\mathbf{s}_1$, consistent with our theory that a desirable disentanglement is attainable.

4.3 Hierarchical Model Identification

Synthetic data. We present the evaluation of Algorithm 1 on latent hierarchical models in Figure 4, Table 2, and Table 4- 5 in Appendix D.3. In Figure 4, we can observe that all variables can be estimated decently, despite a slight recovery loss from a lower to a higher level. Such loss can be mitigated by more careful tuning of the deep learning model. Table 2 presents the pair-prediction scores within pairs of estimations while learning the V-structure model in Figure 4b. We can observe that scores within the sibling pairs $(\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{x}_4, \mathbf{x}_5)$ are much higher than non-sibling pairs.

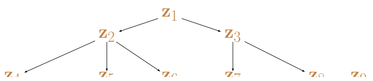
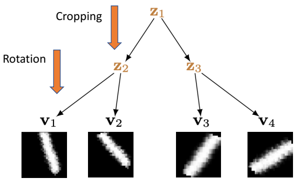


Figure 5: **The causal structure of the personality dataset learned by our method**. The Letter in each variable name indicates the personality attribute. Subscripts correspond to question indices. Some observed variables are not shown

hey are not clustered with other variables,
istant relation to the system.



(a) **The learned causal structure.**

	$\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
$\mathbf{v}_1$	×	$\mathbf{0.69} \pm 0.001$	0.51 ± 0.01	0.51 ± 0.005
$\mathbf{v}_2$	$\mathbf{0.65} \pm 0.02$	×	0.48 ± 0.01	0.47 ± 0.03
$\mathbf{v}_3$	0.48 ± 0.02	0.42 ± 0.03	×	$\mathbf{0.78} \pm 0.05$
$\mathbf{v}_4$	0.53 ± 0.08	0.48 ± 0.02	$\mathbf{0.75} \pm 0.05$	×

(b) **Pairwise predictions.**

Figure 6: **Multi-view digit dataset results.** Figure 6a: $(\mathbf{v}_1, \mathbf{v}_2)$ and $(\mathbf{v}_3, \mathbf{v}_4)$ form two clusters, as they share the aspect-ratio factor within the cluster while distinct in the rotation-angle factor. Table 6b: each box $(\mathbf{a}, \mathbf{b})$ shows the R^2 score obtained by applying the estimated variable produced by treating one specific view $\mathbf{a}$ as $\mathbf{v}_1$ to predict the estimated variable produced by treating view $\mathbf{b}$ as $\mathbf{v}_1$.

Notably, the estimate from $\mathbf{v}_1 = \mathbf{x}_3$ can perform accurate prediction over other estimates, whereas the other estimates fail to capture it faithfully. This is consistent with Theorem 3.4: the basis model with $\mathbf{v}_1 = \mathbf{x}_3$ will output an variable equivalent to the concatenation of $\mathbf{z}_2$ and $\mathbf{z}_3$, which explains the asymmetric prediction performances. These results empirically corroborate Theorem 3.5.

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$
$\mathbf{x}_1$	×	$\mathbf{0.85} \pm 0.001$	0.53 ± 0.03	0.57 ± 0.000	0.55 ± 0.001
$\mathbf{x}_2$	$\mathbf{0.83} \pm 0.002$	×	0.52 ± 0.001	0.54 ± 0.01	0.53 ± 0.000
$\mathbf{x}_3$	$\mathbf{0.90} \pm 0.000$	$\mathbf{0.88} \pm 0.001$	×	$\mathbf{0.86} \pm 0.001$	$\mathbf{0.90} \pm 0.000$
$\mathbf{x}_4$	0.57 ± 0.002	0.54 ± 0.001	0.55 ± 0.001	×	$\mathbf{0.86} \pm 0.001$
$\mathbf{x}_5$	0.55 ± 0.003	0.56 ± 0.003	0.55 ± 0.001	$\mathbf{0.83} \pm 0.003$	×

Table 2: **Pairwise predictions among estimated variables in Figure 4b**. Each box $(\mathbf{a}, \mathbf{b})$ shows the R^2 score obtained applying the estimated variable produced by treating $\mathbf{a}$ as $\mathbf{v}_1$ to predict that produced by treating $\mathbf{b}$ as $\mathbf{v}_1$. We observe that the prediction scores within sibling pairs are noticeably higher than other pairs, suggesting a decent structure estimation. In particular, the estimate from $\mathbf{v}_1 = \mathbf{x}_3$ can predict other estimates accurately, whereas not the other way round, confirming our theory that $\mathbf{v}_1 = \mathbf{x}_3$ will recover the information of both $\mathbf{z}_2$ and $\mathbf{z}_3$.

Personality dataset. From Figure 5, we can observe that our nonlinear model can correctly cluster the variables associated with the same attribute together in the first level, which is consistent with the intentions of these questions. The two sets of results are consistent, with the second set revealing more detailed information about the data – it suggests that conscientiousness, agreeableness, and neuroticism are closely related at the intermediate level, whereas extraversion and neuroticism are more closely related. Some observed variables are not shown in the figure, as they are not clustered with other variables, suggesting they are not closely related to the system. This may provide insights into the efficacy of the design of certain questions.

Digit dataset. Figure 6a and Table 6b present the causal structure learned from the multi-view digit dataset. We can observe that we can automatically cluster the two views sharing more latent factors. This showcases that our theory and approach can handle high-dimensional variables, whereas prior causal structure learning work mostly assumes that all variables are one-dimensional.

5 Conclusion

In this work, we investigate the identifiability of causal structures and latent variables in nonlinear latent hierarchical models. We provide identifiability theory for both the causal structures and latent variables without assuming linearity/discreteness as in prior work [25, 9, 15, 35] while admitting structures more general than (generalized) latent trees [25, 9, 15]. Together with the theory, we devise an identification algorithm and validate it across multiple synthetic and real-world datasets.

In future research, we aim to integrate our theoretical insights into scalable continuous-optimization-based algorithms [39] and deep learning. We believe that our work facilitates the understanding of the underlying structure of highly complex and high-dimensional datasets, which is the foundation for creating more interpretable, safer, and principled machine learning systems.

References

[1] J. Adams, N. Hansen, and K. Zhang. Identification of partially observed linear causal models: Graphical conditions for the non-gaussian and heterogeneous cases. *Advances in Neural Information Processing Systems*, 34, 2021.

[2] S. Akbari, E. Mokhtarian, A. Ghassami, and N. Kiyavash. Recursive causal structure learning in the presence of latent variables and selection bias. In *Advances in Neural Information Processing Systems*, volume 34, pages 10119–10130, 2021.

[3] A. Anandkumar, D. Hsu, A. Javanmard, and S. Kakade. Learning linear bayesian networks with latent variables. In *International Conference on Machine Learning*, pages 249–257, 2013.

[4] R. Cai, F. Xie, C. Glymour, Z. Hao, and K. Zhang. Triad constraints for learning causal structure of latent variables. In *Advances in Neural Information Processing Systems*, pages 12863–12872, 2019.

[5] M. A. Casey and A. Westner. Separation of mixed audio sources by independent subspace analysis. In *ICMC*, pages 154–161, 2000.

[6] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

[7] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky. Latent variable graphical model selection via convex optimization. *Annals of Statistics*, 40(4):1935–1967, 2012.

[8] Z. Chen, F. Xie, J. Qiao, Z. Hao, K. Zhang, and R. Cai. Identification of linear latent variable model with arbitrary distribution. In *Proceedings 36th AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

[9] M. J. Choi, V. Y. Tan, A. Anandkumar, and A. S. Willsky. Learning latent tree graphical models. *Journal of Machine Learning Research*, 12:1771–1812, 2011.

[10] D. Colombo, M. H. Maathuis, M. Kalisch, and T. S. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pages 294–321, 2012.

[11] R. Cui, P. Groot, M. Schauer, and T. Heskes. Learning the causal structure of copula models with latent variables. In *UAI*, pages 188–197. AUAI Press, 2018.

[12] L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[13] M. Drton, S. Lin, L. Weihs, and P. Zwiernik. Marginal likelihood and model selection for gaussian latent tree and forest models. *Bernoulli*, 23(2):1202–1232, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] B. Huang, C. J. H. Low, F. Xie, C. Glymour, and K. Zhang. Latent hierarchical causal structure discovery with rank constraints. *arXiv preprint arXiv:2210.01798*, 2022.

[16] F. Huang, N. U. Naresh, I. Perros, R. Chen, J. Sun, and A. Anandkumar. Guaranteed scalable learning of latent tree models. In *Uncertainty in Artificial Intelligence*, pages 883–893. PMLR, 2020.

[17] A. Hyvärinen and P. Hoyer. Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural computation*, 12(7):1705–1720, 2000.

[18] A. Hyvarinen, H. Sasaki, and R. Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR, 2019.

[19] I. Khemakhem, D. Kingma, R. Monti, and A. Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

[20] B. Kivva, G. Rajendran, P. Ravikumar, and B. Aragam. Learning latent causal graphs via mixture oracles. *Advances in Neural Information Processing Systems*, 34, 2021.

[21] L. Kong, S. Xie, W. Yao, Y. Zheng, G. Chen, P. Stojanov, V. Akinwande, and K. Zhang. Partial disentanglement for domain adaptation. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 11455–11472. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/kong22a.html`.

[22] E. Kummerfeld and J. Ramsey. Causal clustering for 1-factor measurement models. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1655–1664. ACM, 2016.

[23] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR 2011*, pages 3361–3368. IEEE, 2011.

[24] Q. Lyu, X. Fu, W. Wang, and S. Lu. Understanding latent correlation-based multiview learning and self-supervision: An identifiability perspective. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=5FUq05QRc5b`.

[25] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[26] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, NY, USA, 2000. ISBN 0-521-77362-8.

[27] O.-S. P. Project. Big five personality test, 2019. URL `https://openpsychometrics.org/tests/IPIP-BFFM/`. Accessed: 2023-5-16.

[28] S. Shimizu, P. O. Hoyer, and A. Hyvärinen. Estimation of linear non-gaussian acyclic models for latent factors. *Neurocomputing*, 72(7-9):2024–2027, 2009.

[29] R. Silva, R. Scheine, C. Glymour, and P. Spirtes. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7(Feb):191–246, 2006.

[30] P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT press, 2000.

[31] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, 2nd edition, 2001.

[32] F. Theis. Towards a general independent subspace analysis. *Advances in Neural Information Processing Systems*, 19:1361–1368, 2006.

[33] J. von Kügelgen, Y. Sharma, L. Gresele, W. Brendel, B. Schölkopf, M. Besserve, and F. Locatello. Self-supervised learning with data augmentations provably isolates content from style. *arXiv preprint arXiv:2106.04619*, 2021.

[34] F. Xie, R. Cai, B. Huang, C. Glymour, Z. Hao, and K. Zhang. Generalized independent noise condition for estimating latent variable causal graphs. In *Advances in Neural Information Processing Systems*, pages 14891–14902, 2020.

[35] F. Xie, B. Huang, Z. Chen, Y. He, Z. Geng, and K. Zhang. Identification of linear non-gaussian latent hierarchical structure. In *International Conference on Machine Learning*, pages 24370–24387. PMLR, 2022.

[36] W. Yao, Y. Sun, A. Ho, C. Sun, and K. Zhang. Learning temporally causal latent processes from general temporal data. *arXiv preprint arXiv:2110.05428*, 2021.

[37] W. Yao, G. Chen, and K. Zhang. Learning latent causal dynamics. *arXiv preprint arXiv:2202.04828*, 2022.

[38] N. L. Zhang. Hierarchical latent class models for cluster analysis. *The Journal of Machine Learning Research*, 5:697–723, 2004.

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

[40] C. Zhou, X. Wang, and J. Guo. Learning mixed latent tree models. *Journal of Machine Learning Research*, 21(249):1–35, 2020.

Appendix for “Identification of Nonlinear Latent Hierarchical Models"

A Detailed Literature Review

Previous causal discovery approaches, which allow latent confounders and causal relationships among those latent variables, either assume linear causal relationships or assume discrete data. Representative approaches along this line include rank deficiency-based methods, matrix decomposition-based methods, generalized independent noise condition-based methods, and mixture oracles-based methods. (1) Rank deficiency-based. By testing the rank deficiency over cross-covariance matrices over observed variables, one is able to locate latent variables and identify the causal relationships among them in linear-Gaussian models. Silva et al. [29], Kummerfeld and Ramsey [22] make use of the Tetrad condition, a special case of the rank-deficiency constraints, to handle the case where each observed variable is influenced by only one latent parent, and each latent variable has at least three pure measured children. The Tetrad condition has also been used to identify a latent tree structure [25]. Recently, the rank-deficiency constraints have been extended to identify more general hierarchical structures [15]. (2) Matrix decomposition-based. It has been shown that, under certain conditions, the precision matrix can be decomposed into a low-rank matrix and a sparse matrix, where the low-rank matrix represents the causal structure from latent variables to observed variables and the sparse matrix gives the structural relationships over observed variables. To achieve such decomposition, certain assumptions are imposed on the structure [6, 7, 3], e.g., there should be three times more measured variables than latent variables. (3) Generalized independent noise (GIN) condition-based. The GIN condition is an extension of the independent noise condition in the existence of latent confounders, relying on higher-order statistics to identify latent structures. In particular, Xie et al. [34] proposes a GIN-based approach that allows multiple latent parents behind every pair of observed variables and can identify causal directions among latent variables. Moreover, Adams et al. [1] gives necessary and sufficient structural constraints in the linear, non-Gaussian or heterogeneous case, to identify the latent structures. (4) Mixture oracles-based method-based. Recently, Kivva et al. [20] proposed a mixture oracles-based method to identify the latent variable graph that allows nonlinear causal relationships. However, it requires that the latent variables are discrete and each latent variable has measured variables as children. Thanks to the discreteness assumption, it can handle general DAGs over latent variables. On the other hand, regarding the scenario of latent hierarchical structures, most previous work along this line assumes a tree structure and requires that each variable has only one dimension and that the data is either linear-Gaussian or discrete [25, 38, 9, 13, 16]. In contrast, we address the general nonlinear case with continuous variables. Moreover, our conditions allow for multiple undirected paths between two variables and thus are more general than tree-based assumptions in prior work.

Another related research line is latent-variable identifiability literature. Hyvarinen et al. [18], Khemakhem et al. [19], Kong et al. [21] have shown that with an additional observed variable to modulate latent independent variables, the latent independent variables are identifiable. Recently, Yao et al. [36, 37] allow time-delayed causal relationships among latent variables. However, for time-delayed causal relations, the causal direction is fixed and predefined, and moreover, they assume that all latent variables have measured variables as children, avoiding the hierarchical cases. Prior work[33, 24] studies latent-variable models related to our proposed basis model (which serves as a tool and is defined in Section 2) in this work, but with more restrictive functional and statistical assumptions. To the best of our knowledge, no prior work has managed to identify latent variables or causal structures in nonlinear latent hierarchical models.

B Proofs

B.1 Proof for Theorem 3.2

The original proof is replicated here for reference.

In Theorem 3.2, we show that the latent variable $\mathbf{z}$ shared by $\mathbf{v}_1$ and $\mathbf{v}_2$ is identifiable up to an one-to-one mapping by estimating a generative model $(\hat{p}_{\mathbf{z},\mathbf{s}_2}, \hat{p}_{\mathbf{s}_1}, \hat{g})$ according to Equation 2. We denote the support of Jacobian matrix $\mathbf{J}_g$ and $\mathbf{J}_{\hat{g}}$ as $\mathcal{G}$ and $\hat{\mathcal{G}}$ respectively and denote by $\mathbf{T}$ a matrix with the same support as $\mathbf{T}(\mathbf{c})$ in $\mathbf{J}_{\hat{g}}(\hat{\mathbf{c}}) = \mathbf{J}_g(\mathbf{c})\mathbf{T}(\mathbf{c})$.

Assumption 3.1 (Basis model conditions)**.**

i *[Invertibility]: The joint mapping $g(\mathbf{c}) = (\mathbf{v}_1, \mathbf{v}_2)$ is invertible.*

ii *[Subspace span]: For all $i \in \{1, \dots, d_{v_1} + d_{v_2}\}$, there exists $\{\mathbf{c}^{(\ell)}\}_{\ell=1}^{|\mathcal{G}_{i,:}|}$, such that span$(\{\mathbf{J}_g(\mathbf{c}^{(\ell)})_{i,:}\}_{\ell=1}^{|\mathcal{G}_{i,:}|}) = \mathbb{R}^{d_c}_{\mathcal{G}_{i,:}}$ and $[\mathbf{J}_g(\mathbf{c}^{(\ell)})\mathbf{T}]_{i,:} \in \mathbb{R}^{d_c}_{\hat{\mathcal{G}}_{i,:}}$.*

iii *[Edge Connectivity]: For all $j_z \in \{1, \dots, d_z\}$, there exist i_{v_1} and i_{v_2}, such that $(i_{v_1}, j_z) \in \mathcal{G}$ and $(i_{v_2}, j_z) \in \mathcal{G}$.*

Theorem 3.2. *Under Assumption 3.1, if a generative model $(\hat{p}_{\mathbf{z},\mathbf{s}_2}, \hat{p}_{\mathbf{s}_1}, \hat{g})$ follows the data-generating process in Equation 2 and matches the true joint distribution:*

$$p_{\mathbf{v}_1,\mathbf{v}_2}(\mathbf{v}_1, \mathbf{v}_2) = \hat{p}_{\mathbf{v}_1,\mathbf{v}_2}(\mathbf{v}_1, \mathbf{v}_2), \forall(\mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V} \times \mathcal{V}, \tag{3}$$

then the estimated variable $\hat{\mathbf{z}}$ and the true variable $\mathbf{z}$ are equivalent up to a one-to-one mapping.

Proof. We first define the indeterminacy function:

$$h := \hat{g}^{-1} \circ g,$$

which is a smooth and invertible function $h : \mathcal{C} \to \hat{\mathcal{C}}$ thanks to Assumption 3.1-i. According to the chain rule and inverse function theorem, we have the following relation among the Jacobian matrices:

$$\mathbf{J}_{\hat{g}}(\hat{\mathbf{c}}) = \mathbf{J}_g(\mathbf{c}) \cdot \mathbf{J}_h^{-1}(\mathbf{c}). \tag{4}$$

For ease of exposition, we denote $\mathbf{M}(\mathbf{c}) := \mathbf{J}_h^{-1}(\mathbf{c})$ in the following.

We define the support notations as follows:

$$\mathcal{G} := \text{supp}(\mathbf{J}_g),$$
$$\hat{\mathcal{G}} := \text{supp}(\mathbf{J}_{\hat{g}}),$$
$$\mathcal{T} := \text{supp}(\mathbf{M}).$$

Because of Assumption 3.1-ii, for any $i \in \{1, \dots, d_{v_1} + d_{v_2}\}$, there exists $\{\mathbf{c}^{(\ell)}\}_{\ell=1}^{|\mathcal{G}_{i,:}|}$, such that span$(\{\mathbf{J}_g(\mathbf{c}^{(\ell)})_{i,:}\}_{\ell=1}^{|\mathcal{G}_{i,:}|}) = \mathbb{R}^{d_z}_{\mathcal{G}_{i,:}}$.

Since $\{\mathbf{J}_g(\mathbf{c}^{(\ell)})_{i,:}\}_{\ell=1}^{|\mathcal{G}_{i,:}|}$ forms a basis of $\mathbb{R}^{d_c}_{\mathcal{G}_{i,:}}$, for any $j_0 \in \mathcal{G}_{i,:}$, we can express canonical basis vector $\mathbf{e}_{j_0} \in \mathbb{R}^{d_c}_{\mathcal{G}_{i,:}}$ as:

$$\mathbf{e}_{j_0} = \sum_{\ell \in \mathcal{G}_{i,:}} \alpha_\ell \cdot \mathbf{J}_g(\mathbf{c}^{(\ell)})_{i,:}, \tag{5}$$

where $\alpha_\ell \in \mathbb{R}$ is a coefficient.

Then, following Assumption 3.1-ii, there exists a deterministic matrix $\mathbf{T}$ such that

$$\mathbf{T}_{j_0,:} = \mathbf{e}_{j_0}^\top \mathbf{T} = \sum_{\ell \in \mathcal{G}_{i,:}} \alpha_\ell \cdot \mathbf{J}_g(\mathbf{c}^{(\ell)})_{i,:}\mathbf{T} \in \mathbb{R}^{d_c}_{\hat{\mathcal{G}}_{i,:}}, \tag{6}$$

where $\in$ is due to the fact that each element in the summation belongs to $\mathbb{R}^{d_c}_{\hat{\mathcal{G}}_{i,:}}$.

Therefore,

$$\forall j \in \mathcal{G}_{i,:}, \mathbf{T}_{j,:} \in \mathbb{R}^{d_c}_{\hat{\mathcal{G}}_{i,:}}.$$

Equivalently, we have:

$$\forall (i,j) \in \mathcal{G}, \quad \{i\} \times \mathcal{T}_{j,:} \subset \hat{\mathcal{G}}. \tag{7}$$

We would like to show that $\hat{\mathbf{z}}$ is not influenced by $\mathbf{s}_1$ and $\mathbf{s}_2$, which is equivalent to $\mathbf{T}_{j_z,j_{\hat{s}}} = 0$ for $j_z \in \{1, \dots, d_z\}$ and $j_{\hat{s}} \in \{d_z + 1, \dots, d_c\}$.

We first will prove this for $j_{\hat{s}} \in \{d_z+1, \dots, d_z+d_{s_1}\}$ by contradiction. Suppose that $\exists (j_z, j_{\hat{s}_1}) \in \mathcal{T}$ with $j_z \in \{1, \dots, d_z\}$ and $j_{\hat{s}_1} \in \{d_z+1, \dots, d_z+d_{s_1}\}$.

Thanks to Assumption 3.1-iii, there must exist $i_{v_2} \in \{d_{v_1}+1, \dots, d_{v_1}+d_{v_2}\}$, such that, $(i_{v_2}, j_z) \in \mathcal{G}$. It follows from Equation 7 that:

$$\{i_{v_2}\} \times \mathcal{T}_{j_z,:} \subset \hat{\mathcal{G}} \implies (i_{v_2}, j_{\hat{s}_1}) \in \hat{\mathcal{G}}. \tag{8}$$

However, due to the structure of $\hat{g}_2$, $[\mathbf{J}_{\hat{g}_2}]_{i_{v_2}, j_{\hat{s}_1}} = 0$, which results in a contradiction. Therefore, such $(i_{v_2}, j_{\hat{s}_1})$ does not exist. The same reasoning gives rise to that $(j_z, j_{\hat{s}_2}) \notin \mathcal{T}$ with $j_z \in \{1, \dots, d_z\}$ and $j_{\hat{s}_2} \in \{d_z+d_{s_1}+1, \dots, d_c\}$. Therefore, we have shown that $\mathbf{T}_{j_z, j_{\hat{s}}} = 0$ for $j_z \in \{1, \dots, d_z\}$ and $j_{\hat{s}} \in \{d_z+1, \dots, d_c\}$. As $\mathcal{T}$ is invertible, it follows from the block-matrix inverse formulae that $\mathbf{T}_{j_{\hat{z}}, j_s} = 0$ for $j_{\hat{z}} \in \{1, \dots, d_z\}$ and $j_s \in \{d_z+1, \dots, d_c\}$ In conclusion, $\hat{\mathbf{z}}$ is not influenced by $(\mathbf{s}_1, \mathbf{s}_2)$. Thus, we have shown that there is a one-to-one mapping between $\mathbf{z}$ and $\hat{\mathbf{z}}$. □

B.2 Proof for Theorem 3.4

The original theorem is copied below.

Assumption 3.3 (Hierarchical model conditions)**.**

i *[Information-conservation]: Any $\mathbf{z} \in \mathbf{Z}$ and exogenous variable $\boldsymbol{\varepsilon}$ can be expressed as functions of all observed variables, i.e., $\mathbf{z} = f_z(\mathbf{X})$ and $\boldsymbol{\varepsilon} = f_{\varepsilon}(\mathbf{X})$.*

ii *[Subspace span]: For each variable $\mathbf{z}$ and any set $\mathbf{A_z} \subset \mathbf{Y}$ such that 1) $\mathbf{z} \perp\!\!\!\perp \mathbf{X} | \mathbf{A_z}$ and 2) $\mathbf{A_z}$ does not contain directed paths, any invertible function $\tilde{g} : \mathbb{R}^m \to \mathbb{R}^n$ with $\mathbf{z}$ included in its input and $\mathbf{A_z}$ as its output satisfies the subspace span condition (i.e., Assumption 3.1-ii).*

iii *[Edge connectivity]: The function between each latent variable $\mathbf{z}$ and each of its children $\mathbf{z}'$ has a Jacobian $\mathbf{J}$, such that for all $j \in \{1, \dots, d_z\}$, there exists $i \in \{1, \dots, d_{z'}\}$, such that (i, j) is in the support of $\mathbf{J}$.*

Theorem 3.4. *In a latent hierarchical causal model that satisfies Condition 2.3 and Assumption 3.3, we consider $\mathbf{x}_i \in \mathbf{X}$ as $\mathbf{v}_1$ and $\mathbf{X} \backslash \mathbf{v}_1$ as $\mathbf{v}_2$ in the basis model (Figure 2).* [3] *With an estimation model $(\hat{p}_{\mathbf{z}, \mathbf{s}_2}, \hat{p}_{\mathbf{s}_1}, \hat{g})$ that follows the data-generating process in Equation 2, the estimated $\hat{\mathbf{z}}$ is a one-to-one mapping of the parent(s) of $\mathbf{v}_1$, i.e., $\hat{\mathbf{z}} = h(Pa(\mathbf{v}_1))$ where $h(\cdot)$ is an invertible function.*

Proof. We show that performing estimation following the generating process Equation 2 to any $\mathbf{v}_1 = \mathbf{x}_i \in \mathbf{X}$ and $\mathbf{v}_2 = \mathbf{X} \setminus \mathbf{v}_1$ is equivalent to the estimation of the model in Figure 2 with $\mathbf{z} = \text{Pa}(\mathbf{v}_1)$. Therefore, the identifiability result ensues, thanks to Theorem 3.2.

First, we show that for each selection of $\mathbf{v}_1$, we can locate $(\mathbf{z}, \mathbf{s}_1, \mathbf{s}_2)$ such that the conditions for the basis model in Theorem 3.2 are satisfied. We choose $(\mathbf{z}, \mathbf{s}_1, \mathbf{s}_2)$ as follows.

- We choose the $\mathbf{z}$ to be all parents of $\mathbf{v}_1$: $\mathbf{z} := \text{Pa}(\mathbf{v}_1)$
- We choose $\mathbf{s}_1$ to be exogenous variables that cause $\mathbf{v}_1$ together with $\mathbf{z}$: $\mathbf{s}_1 := \boldsymbol{\varepsilon}_{\mathbf{v}_1}$.
- To obtain $\mathbf{s}_2$, we trace back (traversing backward every edge) from the $\mathbf{v}_2$ recursively and execute the following steps. At each step of backtracking, we include any cause (exogenous variables and endogenous variables) that is not situated on any directed path from $\mathbf{z}$ to $\mathbf{v}_2$. The backtracking for each path halts when the most recent step recovers an endogenous variable out of the directed paths, or it reaches $\mathbf{z}$. We note that such a choice for $\mathbf{s}_2$ is not unique – the above procedure is only one instance.

Invertibility. From $(\mathbf{z}, \mathbf{s}_1, \mathbf{s}_2)$ to $(\mathbf{v}_1, \mathbf{v}_2)$, we can observe that $(\mathbf{z}, \mathbf{s}_1, \mathbf{s}_2)$ by construction include all the information to generate $(\mathbf{v}_1, \mathbf{v}_2)$, i.e., $\mathbf{X}$, as $(\mathbf{z}, \mathbf{s}_1, \mathbf{s}_2)$ d-separate their parents/ancestors from $(\mathbf{v}_1, \mathbf{v}_2)$ and contain all necessary exogenous variables. From $(\mathbf{v}_1, \mathbf{v}_2)$ to $(\mathbf{z}, \mathbf{s}_1, \mathbf{s}_2)$, we can observe that as the tuple $(\mathbf{v}_1, \mathbf{v}_2)$ comprises the entire observable set $\mathbf{X}$, which contains the information of all latent exogenous and endogenous variables according to the general invertibility (Assumption 3.3-i) of the hierarchical model. Therefore, the mapping is invertible.

[3] To avoid cluttering, we slightly abuse the bold lowercase font to represent either an individual vector or a vector set (which can be viewed as a concatenation).

Conditional independence: $\mathbf{v}_1 \perp\!\!\!\perp \mathbf{v}_2|\mathbf{z}$**.** As $\mathbf{z}$ is chosen to be the parents of $\mathbf{v}_1$ and there is no edge among the observables $\mathbf{X}$ (i.e., leaf variables), the local Markov property implies conditional independence.

Subspace span. As $(\mathbf{v}_1, \mathbf{v}_2) := \mathbf{X}$ satisfies the tree conditions in Assumption 3.3-ii w.r.t., $\mathbf{z} := \text{Pa}(\mathbf{z})$, the subspace span condition follows from Assumption 3.3-ii.

Edge connectivity. We assume that in the hierarchical model, the function between each latent variable $\mathbf{z}$ and each of its pure child $\mathbf{z}'$ has a non-degenerate Jacobian matrix in a sense that for all $j \in \{1, \ldots, d_z\}$, there exists $i \in \{1, \ldots, d_{z'}\}$, such that (i, j) is included in the Jacobian matrix's support (i.e., Assumption 3.3-iii). Therefore, since each latent variable has at least 2 pure children (i.e., Condition 2.3-i) and $\mathbf{v}_1$ contains 1 variable, at least 1 pure child of $\mathbf{z}$ or a descendant of a pure child of $\mathbf{z}$ will appear in $\mathbf{v}_2$. Thus, for each $j \in \{1, \ldots, d_c\}$, there exist i_{v_1} and i_{v_2} such that both (i_{v_1}, j) and (i_{v_2}, j) are contained in the support of the Jacobian matrix, which fulfills the edge connectivity condition (i.e., Assumption 3.1-iii).

Thus, it follows from Theorem 3.2 that $\hat{\mathbf{z}}$ is a one-to-one mapping to the true $\mathbf{z}$.

□

B.3 Proof for Theorem 3.5

Theorem 3.5. *Under assumptions in Theorem 3.4, all latent variables $\mathbf{Z}$ in the hierarchical model can be identified up to one-to-one mappings, and the causal structure $\mathbf{G}$ can also be identified.*

Proof. We will show by induction that each iteration of Algorithm 1 fulfills the following conditions:

Condition B.1 (Active-set conditions)**.**

i *Each element in the active set $\mathbf{A}$ is a one-to-one mapping of a distinct variable in $\mathbf{Y}$.*

ii *There are no directed paths among variables in $\mathbf{A}$.*

iii *The graph is d-separated by $\mathbf{A}$ into latent variables $\mathbf{Z}_{\mathbf{A}}$ that have not been included in $\mathbf{A}$ and those $\mathbf{Z}_{\bar{\mathbf{A}}}$ that were in $\mathbf{A}$ (but not now): $\mathbf{Z}_{\mathbf{A}} \perp\!\!\!\perp \mathbf{Z}_{\bar{\mathbf{A}}}|\mathbf{A}$.*

We will first verify the base case where active set $\mathbf{A}$ is assigned observable set $\mathbf{X}$. Condition B.1-i is automatically satisfied due to the initial assignment. As $\mathbf{X}$ are all leaf variables, there are no directed paths within $\mathbf{X}$, and therefore Condition B.1-ii is satisfied. Condition B.1-iii is also met trivially, as $\mathbf{Z}_{\bar{\mathbf{A}}} = \emptyset$ at the initial step. So far, we have verified the base case.

Now, we make the inductive hypothesis that all conditions hold before an iteration and show that these conditions are maintained after the iteration.

We first note that Condition B.1-i, Condition B.1-ii, and Condition B.1-iii in the inductive hypothesis enable us to apply Theorem 3.4 to the modified graph consisting of variables in $\mathbf{A}$ as the bottom layer and all latent variables that have not been placed in $\mathbf{A}$. This modified graph satisfies the structural properties required by Theorem 3.4:

- All paths end at active (observed) variables.
- There are no directed paths among the active variables.

We now analyze the updates to $\mathbf{A}$ made at Step 16 in Algorithm 1 after each iteration. For a specific variable $\mathbf{z} \in \mathbf{P}(\mathbf{A})$, there are the following cases.

Case B.2 (One-iteration cases)**.**

i $\mathbf{z}$ *has only pure children and all of them are in $\mathbf{A}$.*

ii $\mathbf{z}$ *has only pure children and not all of them are in $\mathbf{A}$.*

iii $\mathbf{z}$ *has coparents and all its children and its co-parents' children are in $\mathbf{A}$.*

iv $\mathbf{z}$ *has coparents and not all its children and its co-parents' children are in* $\mathbf{A}$.

For Case B.2-i, when each pure child $\mathbf{a}$ of $\mathbf{z}$ is treated as $\mathbf{v}_1$, the basis model will yield $\mathbf{z}$. So, there are certainly estimates of $\mathbf{z}$ in $\mathbf{P}(\mathbf{a})$. As $\mathbf{z}$ possesses at least 2 pure children all of which belong to $\mathbf{A}$, there will be duplicates, i.e., multiple equivalent one-to-one mappings of $\mathbf{z}$, which we merge into one at Step 6. Therefore, for each pure child $\mathbf{a}$ of $\mathbf{z}$, the only element in $\mathbf{P}(\mathbf{a})$ is a identical one-to-one mapping of $\mathbf{z}$. It follows that **JointP** contains a pair $(\mathbf{Z}, \mathbf{H})$ where $\mathbf{Z}$ consists of $\mathbf{z}$ alone and $\mathbf{H}$ contains all its pure children. This fact, together with Condition B.1-iii, implies that $\hat{\mathbf{z}}_{\text{test}}$ would be the parent of $\mathbf{z}$, which cannot perfectly predict $\mathbf{z}$ at Step 14. Therefore, Algorithm 1 will substitute all the pure children of $\mathbf{z}$ with the estimate of $\mathbf{z}$ in $\mathbf{A}$.

We now switch to Case B.2-ii. Analogous to the process in Case B.2-i, after Step 6 we will have $\mathbf{P}(\mathbf{a}) = \{\mathbf{z}\}$ for each active pure child. However, as not all pure children of $\mathbf{z}$ are active, **JointP**$(\{\mathbf{z}\})$ does not include all of the pure children of $\mathbf{z}$ and $\mathbf{A} \setminus \mathbf{H}$ certainly contains descendants of $\mathbf{z}$. It follows from Corollary 3.6 that $\hat{\mathbf{z}}_{\text{test}}$ will be a one-to-one mapping of $\mathbf{z}$ and can predict it perfectly at Step 14. Therefore, $\mathbf{z}$ will not be updated to $\mathbf{A}$ at this iteration.

For Case B.2-iii, we will show that the update takes place for $\mathbf{z}$ and its coparents. Without loss of generality, we assume that $\mathbf{z}$ only has one coparent $\mathbf{z}'$. As both $\mathbf{z}$ has multiple pure children (i.e., Condition 2.3-i) and all of them are in $\mathbf{A}$, the dictionary $\mathbf{P}$ will contain $\mathbf{z}$ and $\mathbf{z}'$ in their keys and $\mathbf{P}(\mathbf{z})$ and $\mathbf{P}(\mathbf{z}')$ contain their pure children respectively. Suppose $\mathbf{a}$ is a shared child of $\mathbf{z}$ and $\mathbf{z}'$, then $\mathbf{P}(\mathbf{a})$ would be a singleton set containing a one-to-one mapping of $\mathbf{z}'' := [\mathbf{z}, \mathbf{z}']$, after Step 6. At Step 9, $\mathbf{z}''$ will be recognized as a combination of $\mathbf{z}$ and $\mathbf{z}'$ and replaced with them. It follows that **JointP** will contain a pair $(\mathbf{Z}, \mathbf{H})$ such that $\mathbf{Z} = (\mathbf{z}, \mathbf{z}')$ and $\mathbf{H}$ is composed of all the children of the two coparents. At Step 16, both $\mathbf{z}$ and $\mathbf{z}'$ will be updated into $\mathbf{A}$. Cases with more than one coparents can be dealt with in the same manner.

For Case B.2-iv, we show that the update will not be carried out for $\mathbf{z}$ and its coparents. Following the notation in Case B.2-iii, we assume that $\mathbf{z}$ has a coparent $\mathbf{z}'$, and these two share at least one active child. If neither $\mathbf{z}$ nor $\mathbf{z}'$ has any pure active children, they will be regarded as a supernode $\mathbf{z}'' = [\mathbf{z}, \mathbf{z}']$ and the corresponding $\mathbf{H}$ contains only the shared active children. Then, $\mathbf{A} \setminus \mathbf{H}$ will certainly contain descendants of both $\mathbf{z}$ and $\mathbf{z}'$. Consequently, $\hat{\mathbf{z}}_{\text{test}}$ will be equivalent to the supernode $\mathbf{z}''$ and predicts it perfectly at Step 14. In this case, neither $\mathbf{z}$ and $\mathbf{z}'$ will be updated into $\mathbf{A}$. If only one of $\mathbf{z}$ and $\mathbf{z}'$ has no pure children in $\mathbf{A}$, then neither of them will be updated into $\mathbf{A}$ due to Step 9. Without loss of generality, we assume that $\mathbf{z}'$ does not have any active pure children, whereas $\mathbf{z}$ does. Then, there will be no estimated $\mathbf{z}'$ in the values of $\mathbf{P}$, but there will be estimated $\mathbf{z}$ and $(\mathbf{z}, \mathbf{z}')$. As a results, Step 9 will prevent $\mathbf{z}$ and $\mathbf{z}'$ from updated to $\mathbf{A}$. If both $\mathbf{z}$ and $\mathbf{z}'$ have active pure children, then there will be a pair $(\mathbf{Z}, \mathbf{H})$ in **JointP** such that $\mathbf{Z} = \{\mathbf{z}, \mathbf{z}'\}$ and $\mathbf{H}$ contains all their active children. As $\mathbf{H}$ does not contain all the children of $\mathbf{z}$ and $\mathbf{z}'$, the $\mathbf{A} \setminus \mathbf{H}$ will contain descendants of at least one of $\mathbf{z}$ and $\mathbf{z}'$. Then $\hat{\mathbf{z}}_{\text{test}}$ will contain all information of at least one of $\mathbf{z}$ and $\mathbf{z}'$ and predicts it perfectly at Step 16, which aborts the update of $\mathbf{z}$ and $\mathbf{z}'$. Therefore, no updates will happen in this case.

As each of the newly estimated variables in the values of **JointP** is a one-to-one mapping to a distinct variable in $\mathbf{Z}$, the iteration preserves Condition B.1-i.

For Condition B.1-ii, we can observe that if $\mathbf{z}$ is updated into $\mathbf{A}$ at this iteration, all its children would be in $\mathbf{A}$ at the beginning of this iteration and get removed at Step 16. Therefore, the update would not introduce directed paths to $\mathbf{A}$. Condition B.1-iii holds for the same reason: the newly introduced variables in the active set $\mathbf{A}$ at the end of the iteration separate their children in the active set $\mathbf{A}$ at the beginning of the iteration and all undiscovered latent variables.

So far, by induction, we have shown that Condition B.1-iii, Condition B.1-ii, and Condition B.1-iii hold at the end of each iteration. Given the analysis above, at each iteration, we update variables whenever all their children and their coparents' children are active, i.e., Case B.2-i and Case B.2-iii, and the causal relations between the parents and the children are encoded in **JointP**.

We further argue that at every iteration, there exists at least one undiscovered latent variable that has all its children in $\mathbf{A}$. We focus on the trimmed graph consisting of $\mathbf{Z}_{\mathbf{A}}$ and $\mathbf{A}$, where $\mathbf{Z}_{\mathbf{A}}$ refers to all undiscovered variables. As the graph is of finite size and all variables in $\mathbf{Z}_{\mathbf{A}}$ have directed paths ending at active variables in $\mathbf{A}$, the active variable $\mathbf{a}$ farthest from the root does not have siblings in

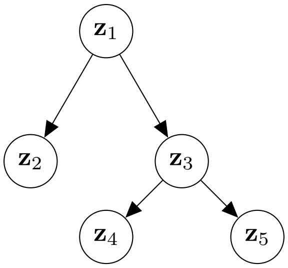


Figure 7: A specific case to demonstrate how we can avoid assuming specific functional classes.

$\mathbf{Z}_\mathbf{A}$; otherwise, there would be $\mathbf{a}'$ (i.e., the child of $\mathbf{a}$'s sibling in $\mathbf{Z}_\mathbf{A}$) in $\mathbf{A}$ that was further from the root than $\mathbf{a}$.

In sum, we have shown that there will exist at least one update at each iteration, and each update will lead to the discovery of new latent variables and correct causal structures. Therefore, Algorithm 1 can successfully identify the underlying hierarchical causal graph and each latent variable up to one-to-one mappings.

□

B.4 Illustration of a Specific Case

Figure 7 shows a specific case to demonstrate how we can avoid assuming specific functional classes. For active set $\mathbf{A} = \{\mathbf{z}_2, \mathbf{z}_4, \mathbf{z}_5\}$, if we treat $\mathbf{z}_2$ as $\mathbf{v}_1$, $\mathbf{z}_1$ will become $\mathbf{z}$ in the basis and $[\mathbf{z}_4, \mathbf{z}_5]$ will become $\mathbf{v}_2$. This set of $(\mathbf{z}, \mathbf{v}_1, \mathbf{v}_2)$ satisfies Assumption 2-i, as all information about $\mathbf{z}_1$ is contained by $\mathbf{z}_2$, $\mathbf{z}_4$, and $\mathbf{z}_5$. Assumption 2-iii and Assumption 2-ii are direct consequences of Assumption 3.3-iii and Assumption 3.3-ii. Also, we can always find an independent variable $\mathbf{s}_1 := \varepsilon_{z_2}$ in the basis model. Therefore, Theorem 3.4 guarantees the estimated $\hat{\mathbf{z}}$ to be an one-to-one mapping of $\mathbf{z}_1$.

C Relaxation of Structural Conditions

C.1 Relaxation of Condition 2.3-ii

We remark that Condition 2.3-ii is introduced to simplify the presentation of our main results – as the first step of nonlinear-latent-hierarchical causal discovery, we wish to present the basic techniques more clearly – handling triangles involves additional components of the algorithm and could potentially compromise the readability.

We give an example in Figure 8 with a triangle structure to illustrate how our theory and algorithm can handle triangles. We refer to variables in a triangle under the following convention: 1) root: the variable which is the cause of the other two variables (e.g., $\mathbf{z}_1$ in Figure 8), 2) sink: the variable which is the effect of the other two variables (e.g., $\mathbf{z}_3$ in Figure 8), and 3) bridge: the variable which is the effect of the root variable and the cause of the sink variables (e.g., $\mathbf{z}_2$ in Figure 8).

In the following, we illustrate the procedures to recognize and handle triangle structure by navigating through Figure 8. First, the directed path detection procedure portrayed in Section 3.3 can pinpoint the directed path between the bridge variable and the sink variable (e.g., $\mathbf{z}_2$ and $\mathbf{z}_3$). We need to determine whether the two variables indeed belong to a triangle. To this end, we can *lump* the bridge variable and the sink variable together as $\mathbf{v}_1$. We perform the basis model estimation over $\mathbf{v}_1$ and the active set $\mathbf{A}$ excluding $\mathbf{v}_1$'s children. In Figure 8, we perform the basis model over $\mathbf{v}_1 = [\mathbf{z}_2, \mathbf{z}_3]$ and $\mathbf{A} = \{\mathbf{z}_2, \mathbf{z}_3, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9\}$, which returns an estimate of $\mathbf{z}_1$. Further, we can determine that the bridge variable $\mathbf{z}_2$ and the estimated variable from the lumped $\mathbf{v}_1$ constitute the parents of the sink variable $\mathbf{z}_3$ by mutual prediction, which ascertains the two variables are indeed the bridge variable and the sink variable of a triangle, respectively. In this example, we find that the estimated variables of $\mathbf{z}_2$ and $\mathbf{z}_1$ together contain identical information to the estimated variable of $\mathbf{z}_3$, which reveals that $\mathbf{z}_2$ and $\mathbf{z}_3$ are the bridge variable and the sink variable of a triangle. This concludes the triangle detection procedure. With this knowledge, we can handle the triangle structure by lumping the bridge variable $\mathbf{z}_3$ and the sink variable $\mathbf{z}_3$ into a super-variable and proceed with this modification.

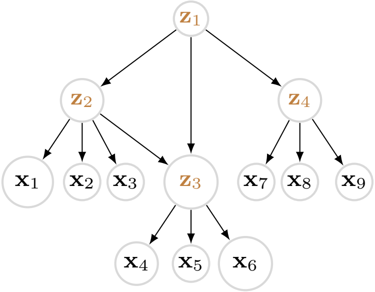


Figure 8: An example of triangle structures, i.e., $\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{z}_3$.

C.2 Admitting Non-leaf variables into $\mathbf{X}$

Identical to the directed path detection detailed in Section 3.3, Corollary 3.6 can detect non-leaf observed variables and exclude them from the initial active set $\mathbf{X}$ in Algorithm 1. Thus, the problem can be reduced back to the case without observed non-leaf variables, which can be tackled by Algorithm 1.

D Experiments

D.1 Additional Experimental Details

We provide additional details of experiments in Section 4. For the basis model evaluation in Section 4.2, then encoder $\hat{f}$ and decoders $(\hat{g}_1, \hat{g}_2)$ are 4-layer MLP's with a hidden dimension 30 times as large as the input data and Leaky-ReLU ($\alpha = 0.2$) activation functions. For the synthetic hierarchy experiments in Section 4.3, the model layers are 8 and 50 times as large as the input data. For the personality dataset in Section 4.3, the model layers are 4 and 8 times as large as the input data. For the multi-view digit dataset in Section 4.3, the model layers are 4, and 4 times (encoder) and 2 times (decoder) as large as the input data The model configurations are summarized in Table 3.

	# encoder layers	# decoder layers	encoder widths	decoder width
Basis models (Section 4.2)	4	4	30×	30×
Synthetic hierarchy (Section 4.3)	8	8	50×	50×
Personality dataset (Section 4.3)	4	4	8×	8×
Digit dataset (Section 4.3)	4	4	4×	2×

Table 3: **Estimation model configuration for each experiment.**

We apply Adam to train each model for $20,000$ steps with a learning rate of $1e-3$. For hierarchical models, we consistently use 2-layer MLPs for generating functions and set the exogenous variable dimensionality as 2. We conduct our experiments on two V-100 Nvidia GPUs. Synthetic data experiments in Section 4.2 and Section 4.3 take around half an hour and two hours for one run, respectively. Personality dataset experiments and digit dataset experiments call for three hours for one round.

D.2 Additional Results on the Basis Model

Figure 9 visualizes the dependence between the estimated components and the true components. We can observe that the $\mathbf{z}$ is highly correlated with $\hat{\mathbf{z}}$ whereas $\mathbf{s}_1$ has little correlation with $\hat{\mathbf{z}}$, which verifies Theorem 3.2 that we can successfully identify $\mathbf{z}$.

D.3 Additional Results on Synthetic Hierarchical Models.

Table 4 and Table 5 contain the mutual prediction scores for the hierarchical structures in Figure 4a and Figure 4c. We can observe that the estimated variables that yield high mutual prediction scores are indeed under the same causal cluster, which gives signals for structure learning.

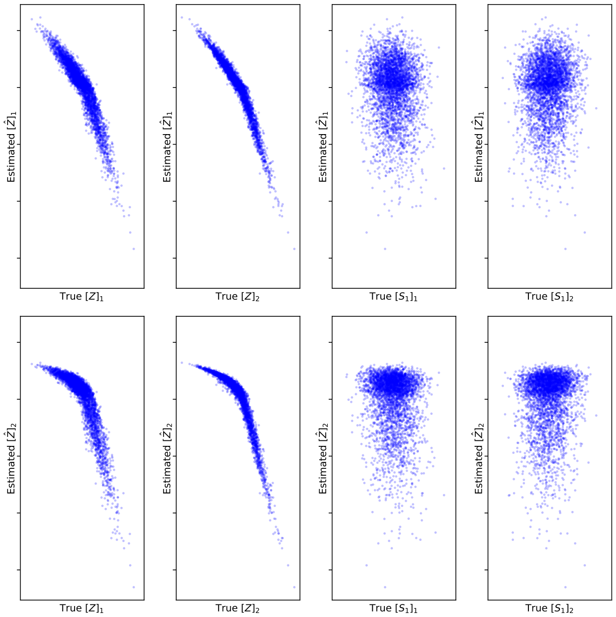


Figure 9: **The scatter plot between the estimated $\hat{\mathbf{z}}$ and the true $(\mathbf{z}, \mathbf{s}_1)$.** The scatter points are obtained in an estimation run with $d_z = d_{s_1} = d_{s_2} = 2$. We can observe that the estimated components of $\hat{\mathbf{z}}$ are highly correlated with those of $\mathbf{z}$ and have little correlation with those $\mathbf{s}_1$, empirically verifying Theorem 3.2.

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$
$\mathbf{x}_1$	×	$\mathbf{0.83} \pm 0.03$	0.57 ± 0.04	0.50 ± 0.08
$\mathbf{x}_2$	$\mathbf{0.86} \pm 0.01$	×	0.58 ± 0.03	0.52 ± 0.04
$\mathbf{x}_3$	0.6 ± 0.02	0.59 ± 0.05	×	$\mathbf{0.77} \pm 0.00$
$\mathbf{x}_4$	0.59 ± 0.02	0.59 ± 0.05	$\mathbf{0.81} \pm 0.04$	×

Table 4: **Pairwise predictions among estimates in Figure 4a**. Each box $(\mathbf{a}, \mathbf{b})$ shows the R^2 score obtained by applying the estimate produced by treating $\mathbf{a}$ as $\mathbf{v}_1$ to predict the estimate produced by treating $\mathbf{b}$ as $\mathbf{v}_1$. We can observe that the prediction scores within sibling pairs $(\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{x}_3, \mathbf{x}_4)$ are noticeably higher than other pairs, showing a decent structure estimation.

D.4 Additional Results on the Personality Dataset

We randomly and evenly partition each set of six questions into two variables. Following this procedure, we have observed ten variables $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{n}_1$, $\mathbf{n}_2$, $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{o}_1$, and $\mathbf{o}_2$, each of 3-dimension. The Letter in each variable name indicates the attribute to which the variable belongs, to facilitate readability. Figure 10 contains the results. Analogous to Figure 5, the learned structure faithfully reflects the questionnaire structure – questions designed for the same personality trait are clustered together.

	$\mathbf{z}_4$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$
$\mathbf{z}_4$	×	$\mathbf{0.81} \pm 0.011$	0.48 ± 0.006	0.52 ± 0.000
$\mathbf{x}_3$	$\mathbf{0.87} \pm 0.002$	×	0.47 ± 0.003	0.54 ± 0.002
$\mathbf{x}_4$	0.57 ± 0.000	0.52 ± 0.006	×	$\mathbf{0.77} \pm 0.005$
$\mathbf{x}_5$	0.58 ± 0.001	0.57 ± 0.015	$\mathbf{0.81} \pm 0.000$	×

Table 5: **Pairwise predictions among estimated variables in Figure 4c**. The pattern is consistent with Table 4 and Table 2.

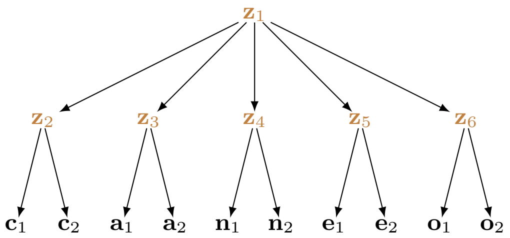


Figure 10: The casual graph learned by our method, where each observed variable ($\mathbf{c}$, $\mathbf{a}$, $\mathbf{n}$, $\mathbf{e}$, $\mathbf{o}$) is of three dimensions.