La Matematica
https://doi.org/10.1007/s44007-025-00168-8

ORIGINAL RESEARCH ARTICLE

Linear Causal Disentanglement via Higher-Order Cumulants

Paula Leyes Carreno[1] · **Chiara Meroni**[2] · **Anna Seigal**[1]

Received: 3 June 2024 / Revised: 18 May 2025 / Accepted: 7 July 2025

Abstract
Linear causal disentanglement (LCD) is a recent method in causal representation learning to describe a collection of observed variables via latent variables with causal dependencies between them. It can be viewed as a generalization of both independent component analysis and linear structural equation models. We study the identifiability of LCD, assuming access to data under multiple contexts, each given by an intervention on a latent variable. We show that one perfect intervention on each latent variable is sufficient and in the worst case necessary to recover parameters under perfect interventions, generalizing previous work to allow more latent than observed variables. We give a constructive proof that computes parameters via a coupled tensor decomposition. For soft interventions, we find the equivalence class of latent graphs and parameters that are consistent with observed data, via the study of a system of polynomial equations. Our results hold assuming the existence of non-zero higher-order cumulants, which implies non-Gaussianity of variables.

Keywords Causal inference · Disentanglement · Higher-order cumulants · Tensor decomposition · Causal representation learning · Interventions

✉ Chiara Meroni
chiara.meroni@eth-its.ethz.ch

Paula Leyes Carreno
pleyescarreno@college.harvard.edu

Anna Seigal
aseigal@seas.harvard.edu

[1] Harvard University, Cambridge, MA, USA

[2] ETH-ITS, Zurich, Switzerland

Published online: 30 September 2025

1 Introduction

A key challenge of data science is to find useful and interpretable ways to model complex data, such as those collected from a biological experiment or a physical system. In this paper, we study *linear causal disentanglement* (LCD), a framework to model such data. LCD generalizes two 20th century data analysis models: *independent component analysis* (ICA) [10, 11, 24] and *linear structural equation models* (LSEMs) [7, 52]. Before defining it, we briefly recall these older models.

ICA is a blind source separation method that expresses observed variables $X = (X_1, \ldots, X_p)$ as a linear mixture

$$X = A\varepsilon, \tag{1}$$

where $A \in \mathbb{R}^{p\times q}$ is a mixing matrix and $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_q)$ is a vector of independent latent variables. ICA has been used in applications including brain dynamics [23] and astrophysics [6]. LSEMs are another linear model to describe collections of variables. They model variables $Z = (Z_1, \ldots, Z_q)$ as

$$Z = \Lambda Z + \varepsilon, \tag{2}$$

where $\Lambda \in \mathbb{R}^{q\times q}$ is a matrix whose entry $\lambda_{i,j}$ encodes the dependence of Z_i on Z_j and ε is a vector of noise variables, often assumed to be independent. The variables are typically assumed to relate via the recursive structure of a directed acyclic graph (DAG); that is, fixing a DAG $\mathcal{G}$ on nodes $[q] = \{1, \ldots, q\}$, with directed edges denoted $j \to i$, we have

$$\lambda_{i,j} \neq 0 \qquad \Longleftrightarrow \qquad (j \to i) \in \mathcal{G}.$$

Equation (2) can be re-written as $Z = (I - \Lambda)^{-1}\varepsilon$, where acyclicity of $\mathcal{G}$ ensures that the matrix $I - \Lambda$ is invertible. This places LSEMs in the context of ICA, since the variables Z are a linear mixing of independent latent variables [45]. LSEMs appear in applications including epidemiology [44] and causal inference [42]. In causal inference, the quantity $\lambda_{i,j}$ is interpreted as the causal effect of Z_j on Z_i.

The idea of linear causal disentanglement [50] is that the assumptions of ICA and LSEMs may be too strict: interpretable latent variables may not be independent, and variables that relate via a graph may not have been directly measured. To get around this, LCD is defined as follows. As in ICA, we observe variables $X = (X_1, \ldots, X_p)$ that are a linear mixing of latent variables. However, unlike ICA, the latent variables are not independent, instead they follow the structure of an LSEM; that is,

$$X = FZ, \qquad \text{where} \qquad Z = \Lambda Z + \varepsilon, \tag{3}$$

for $F \in \mathbb{R}^{p\times q}$ a linear transformation, Λ a matrix that encodes causal dependencies among the latent variables $Z = (Z_1, \ldots, Z_q)$, and $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_q)$ a vector of independent noise variables. As often the case in ICA and LSEMs, variables ε are

assumed to be mean-centered. LCD specializes to ICA when Λ is the zero matrix (i.e. when $\mathcal{G}$ is the empty graph) and to an LSEM when $F = I$.

LCD falls into the setting of *causal representation learning* [46], an area of machine learning that aims to describe and explain the structure of a complex system by learning variables together with the causal dependencies among them. The idea is that learned latent representations of data [5] can be difficult to interpret and analyze, and may not generalize well, but that they improve by using latent representations with causal structure [64]. Central to interpretability and downstream analysis is the identifiability of a representation. The LCD model (3) is identifiable if the mixing matrix F and matrix of dependencies Λ, and therefore also the latent DAG $\mathcal{G}$, can be recovered uniquely (or up to a well-described set of possibilities) from observations of X.

In this paper, we study the identifiability of LCD and develop algorithms to recover the parameters F and Λ using tensor decomposition of higher-order cumulants. Higher-order cumulants have been used to recover parameters in both ICA and LSEMs [11, 15, 45, 59, 60]. We build on these insights to use it for LCD. For ICA and LSEMs, parameters can be recovered from tensor decomposition of a single higher-order cumulant. For LCD one tensor decomposition no longer suffices to recover parameters and we will instead use a coupled tensor decomposition. Identifiability of LCD from covariance matrices (that is, second-order cumulants) was studied in [50]. Our results extend these insights to identifiability via higher-order cumulants.

The setup. Our goal in this paper is to use observations of X to recover the parameters F and Λ in an LCD model (3). We assume access to observations of X under multiple contexts. The contexts differ from an observational context by an intervention. Interventions appear in biological applications such as [16, 39, 48, 53, 62]. Throughout this paper, we assume that the contexts are interventions at a single node. An intervention at a variable affects the downstream variables but not those that are upstream. It thus enables one to find the direction of a causal dependency between two variables. We study multiple contexts for two reasons: inferring causal dependencies in general necessitates interventions and one context is insufficient for recovery of parameters in the model. We consider two types of interventions.

Definition 1.1 Let variables Z_i relate via a linear structural equation model. A *soft intervention* at Z_i changes all non-zero weights $\lambda_{i,j}$ and changes the error distribution ε_i. A *perfect intervention* at Z_i zeros out all non-zero weights $\lambda_{i,j}$ and changes the error distribution ε_i.

A third widely-studied type of intervention is a *do-intervention*, which sets a variable to a deterministic value. We focus on soft interventions and perfect interventions, so that we do not assume access to a fixed value of an unobserved variable. For related results for do-interventions, see [4, 64].

We denote the set of contexts by K, which is assumed to be known. Each context $k \in K$ is assumed to be an intervention at a single latent variable, as in [50]. The target of each intervention is unknown: context k is an intervention on Z_{i_k} for some $i_k \in [q]$. The observational setting, in which no variable is intervened on, is indexed by $k = 0$ and assumed to be known. The intervention changes the latent LSEM but

not the mixing map F. Under context k, we denote the matrix of causal effects by $\Lambda^{(k)}$, the latent variables by $Z^{(k)}$, and the error distributions by $\varepsilon^{(k)}$. Error distributions $\varepsilon^{(k)}$ and $\varepsilon^{(0)}$ agree, except at the i_k-th entry. From Definition 1.1, we see that a perfect intervention sets the i_k-th row of $\Lambda^{(k)}$ to zero while a soft intervention satisfies $\lambda^{(k)}_{i_k,j} \neq \lambda^{(0)}_{i_k,j}$ whenever $\lambda^{(0)}_{i_k,j} \neq 0$, i.e. for all j with edge $j \to i_k$ present in $\mathcal{G}$. Our setup can now be summarized as follows.

Fix $p \geq 2$ observed variables. We observe distributions $X^{(k)}$ on $\mathbb{R}^p$ for $k \in K \cup \{0\}$ of the form

$$X^{(k)} = FZ^{(k)}, \qquad \text{where} \qquad Z^{(k)} = \Lambda^{(k)} Z^{(k)} + \varepsilon^{(k)}, \tag{4}$$

for $Z^{(k)}$ some random variables on $\mathbb{R}^q$ where $q \geq 2$ is the number of latent variables. The variables $Z^{(0)}$ on $\mathbb{R}^q$ follow a linear structural equation model on an unknown DAG $\mathcal{G}$ on q nodes, and $Z^{(k)}$ relates to $Z^{(0)}$ via a single-node perfect or soft intervention with unknown target. See Fig. 1 for a cartoon of our setup. We make the following genericity assumptions.

Assumption 1.2 (a) All noise variables $\varepsilon^{(k)}_i$ are non-Gaussian.

(b) Matrix $F \in \mathbb{R}^{p \times q}$ is unknown and generic; matrices $\Lambda^{(k)} \in \mathbb{R}^{q \times q}$, $k \in K \cup \{0\}$ are unknown with generic non-zero entries.

(c) For all contexts $k \in K$ there exists a large enough d ($d \geq 3(q-1)$ is sufficient) such that the d-th order cumulant of $\varepsilon^{(k)}_{i_k}$ is not 0 or equal to the d-th order cumulant of $\varepsilon^{(0)}_{i_k}$.

Problem 1.3 In the setup (4) under Assumption 1.2, recover the number of latent variables q, the latent DAG $\mathcal{G}$, the mixing matrix F and the matrices of dependencies $\{\Lambda^{(k)} \mid k \in K \cup \{0\}\}$.

We can rearrange (4) to write variables $X^{(k)}$ as a linear mixture of independent latent variables

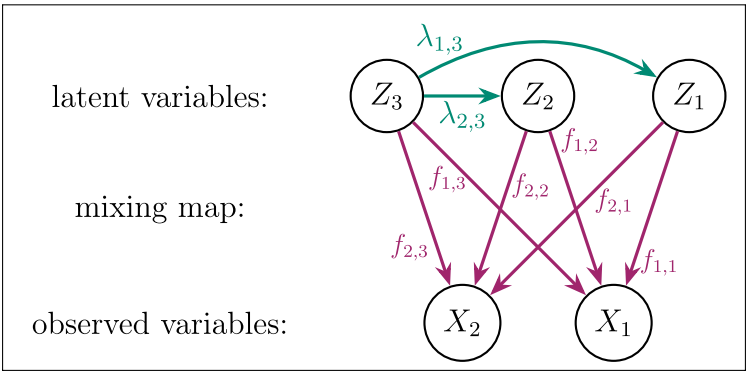


Fig. 1 A cartoon of the setup for $p = 2$ observed variables and $q = 3$ latent variables

$$X^{(k)} = F(I - \Lambda^{(k)})^{-1}\varepsilon^{(k)}.$$

This relates LCD to ICA. Just as for ICA, we have the following non-identifiability.

Remark 1.4 *(Benign non-identifiability)* Uniqueness of F and $\Lambda^{(k)}$ is impossible in LCD, since one can rescale or reorder the latent variables without affecting membership in the model. That is, for a full-rank diagonal matrix $D \in \mathbb{R}^{q\times q}$ and a permutation matrix $P \in \mathbb{R}^{q\times q}$, setting

$$\widetilde{F} = FM, \quad \widetilde{\Lambda}^{(k)} = M^{-1}\Lambda^{(k)}M, \quad \widetilde{\varepsilon}^{(k)} = M^{-1}\varepsilon^{(k)}, \quad \text{where } M = DP, \quad (5)$$

we have

$$\widetilde{F}(I - \widetilde{\Lambda}^{(k)})^{-1}\widetilde{\varepsilon}^{(k)} = F(I - \Lambda^{(k)})^{-1}\varepsilon^{(k)}.$$

Hence such rescaling and reordering does not affect $X^{(k)}$. Such transformations do not change the latent graph $\mathcal{G}$ except by a relabelling of its nodes under the permutation P. Given multiple contexts $k \in K \cup \{0\}$, scaling and ordering transformations D and P are the same for all k.

Definition 1.5 An LCD model is *identifiable* if there exists a DAG $\mathcal{G}$ and matrices F, $\Lambda^{(k)}$ that give the observed distributions $X^{(k)}$ for $k \in K \cup \{0\}$, via the equations in (4), where the matrices F, $\Lambda^{(k)}$ are unique up to the benign rescaling and reordering transformation in (5) and the DAG is unique up to a relabeling of nodes.

Main results.

We find the perfect interventions needed for identifiability of LCD.

Theorem 1.6 *Consider LCD under Assumption 1.2 with perfect interventions. Then one perfect intervention on each latent node is sufficient and, in the worst case, necessary to recover the latent DAG $\mathcal{G}$ and the parameters F and $\Lambda^{(k)}$ from observations of $X^{(k)}$.*

For p observed variables and q latent variables, Theorem 1.6 says that we need q interventions for identifiability of LCD. We do not impose the injectivity of the mixing map $F : \mathbb{R}^q \to \mathbb{R}^p$; the pair (p, q) can take any values provided $p, q \geq 2$. Our proof is constructive: we carry out a coupled tensor decomposition of higher-order cumulants of the distributions $X^{(k)}$, and compare the factors recovered to estimate the parameters. This extends [45, 59] from observed to latent causal variables, and extends [11, 15, 60] from independent to dependent latent variables. It relates to [17], which says that $q - 1$ interventions are sufficient and in the worst case necessary to recover a DAG on q observed variables. It builds on [50, Theorem 1], which says that one intervention on each latent node is sufficient and in the worst case necessary when the mixing F is injective. When the mixing map is injective, Theorem 1.6 is weaker than [50, Theorem 1], since it requires non-Gaussian errors. When F is not injective non-Gaussianity is necessary for identifiability, see Proposition 3.6.

We present two algorithms for the recovery of the model parameters using q perfect interventions. The first algorithm can be used for any (p, q). It takes as input a tuple of $q+1$ cumulants, and returns the parameters F and $\Lambda^{(k)}$. The second algorithm applies to the setting $q \leq p$. Here Moore-Penrose pseudo-inverses can be used to simplify the recovery. We illustrate the performance of the algorithms in Fig. 2. Both are implemented in Python, version 3.12.2. The code is available at:

https://github.com/paulaleyes14/linear-causal-disentanglement-via-cumulants.

We now turn to soft interventions. The transitive closure $\overline{\mathcal{G}}$ of a DAG $\mathcal{G}$ is the DAG with all edges $j \to i$ whenever $j \to \cdots \to i$ is a path in $\mathcal{G}$. We can recover the transitive closure $\overline{\mathcal{G}}$ of a latent DAG $\mathcal{G}$ in LCD from the second-order cumulants, see [50, Theorem 1]. We show that, if the errors are non-Gaussian, we can distinguish certain DAGs with the same transitive closure. We define the set of soft-compatible DAGs $\operatorname{soft}(\mathcal{G})$. It is a set of DAGs with the same transitive closure, which also satisfy additional compatibility conditions coming from ranks of matrices. Define the set of children of node j by $\operatorname{ch}_{\mathcal{G}}(j) = \{i \mid (j \to i) \in \mathcal{G}\}$ and the descendants by $\operatorname{de}_{\mathcal{G}}(j) = \{i \mid (j \to \cdots \to i) \in \mathcal{G}\}$. Then,

$$\operatorname{soft}(\mathcal{G}) = \left\{ \mathcal{G}' \;\middle|\; \overline{\mathcal{G}'} = \overline{\mathcal{G}} \text{ and } \operatorname{rank}[(I - \Lambda_{\mathcal{G}})^{-1}]_{\nu_j, \circ_j} = \operatorname{rank}[(I - \Lambda_{\mathcal{G}})^{-1}]_{\nu_j, \circ_j \cup \{j\}} \text{ for all } j \in [q] \right\},$$

where $\nu_j := \operatorname{de}_{\mathcal{G}'}(j) \setminus \operatorname{ch}_{\mathcal{G}'}(j)$, $\circ_j := \operatorname{ch}_{\mathcal{G}'}(j)$, and $\Lambda_{\mathcal{G}}$ is a generic matrix of dependencies in an LSEM on DAG $\mathcal{G}$, and $[M]_{\nu,\circ}$ denotes the submatrix of M with row indices in ν and column indices in $\circ$. See Definition 3.14 for more details.

Theorem 1.7 *Consider LCD under Assumption 1.2 with soft interventions. Then one soft intervention on each latent node is sufficient and, in the worst case, necessary to*

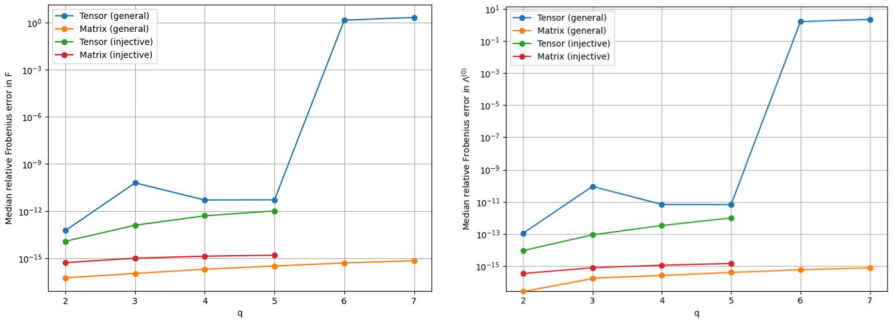


Fig. 2 Median relative Frobenius error in the recovery of F (left) and $\Lambda^{(0)}$ (right) when $p = 5$. Note the logarithmic scale on the y-axis. The four algorithms are: (i) Tensor (general), the general algorithm with cumulants as input (blue), (ii) Matrix (general), the general algorithm with factor matrices as input (orange), (iii) Tensor (injective), the injective algorithm with cumulants as input (green), and (iv) Matrix (injective), the injective algorithm with factor matrices as input. For DAG recovery, all methods recovered the correct DAG every time, except the general tensor method when $q \geq 6$. This had a median DAG recovery error of 3.6 for $q = 6$ and 4.1 for $q = 7$

recover the set of DAGs $\mathrm{soft}(\mathcal{G})$. *Given* $\mathcal{G}' \in \mathrm{soft}(\mathcal{G})$, *the set of parameters* F *and* $\Lambda^{(k)}$ *that are compatible with the observations is a positive dimensional linear space.*

The proof relies on the study of the solution space to a system of polynomial equations, encoding the conditions that parameters compatible with the observations must satisfy. That space is linear and always positive dimensional, even if we allow multiple interventions on each latent node. This leads to a negative identifiability result, in the same spirit as [30].

Corollary 1.8 *Consider LCD under Assumption 1.2. With any number of soft interventions, identifiability of all parameters in the model does not hold.*

The non-Gaussianity assumption is required for the linear space of parameters in Theorem 1.7: with Gaussian errors, the space of parameters may be non-linear, see Proposition 3.3.

Related work. Higher-order cumulants have been shown to lead to improved identifiability in related contexts. They extend principal component analysis, which requires an orthogonal transformation for identifiability, to ICA, which is identifiable for general linear mixings [10, 11]. For LSEMs, they facilitate the recovery of a full DAG, rather than its Markov equivalence class [58], see [45, 59]. They have been used to recover parameters in other latent variable models [1].

Identifiability of causal representation learning is an active area of study. It builds on work in the identifiability of representation learning [2, 25, 66] and latent DAG models. These include work that imposes sparsity on the causal relations [2, 3, 13, 21, 22, 31, 33, 40, 51, 61, 63, 65, 67] and latent variable models on discrete variables [20, 27]. There are many works related to LCD, due in part to the many possible assumptions that one can make in a causal disentanglement model. These include the structure (polynomial, non-linear) of the maps involved [8, 34, 35, 54–57] and the choice of data generating process [9, 28, 32, 47]. In general, allowing more freedom on one side, implies more restrictions on the other side.

Outline We cast LCD as the problem of aligning the outputs of a coupled tensor decomposition in Sect. 2. We discuss the recovery of parameters for perfect and soft interventions in Sect. 3. We prove our main results Theorem 1.6 in Sect. 3.2 and Theorem 1.7 in Sect. 3.3. We discuss our algorithms in Sect. 4 and future directions in Sect. 5. Appendix A contains pseudo-code for our algorithms.

2 Coupled Tensor Decomposition

The cumulants are a sequence of tensors that encode a distribution [38]. The d-th cumulant of a distribution X on $\mathbb{R}^p$ is an order d tensor, denoted by $\kappa_d(X)$, of format $p \times \cdots \times p$. The first and second order cumulants are the mean and covariance, respectively. Higher-order cumulants are those of order three and above.

We describe the higher-order cumulant tensors of distributions $X^{(k)}$ coming from LCD, as in (3), as k ranges over contexts. We study a coupled decomposition of these tensors. This will enable us to study the identifiability of LCD and to design

tensor decompositions to recover parameters in the model. We first consider a single context.

2.1 Decomposing Cumulants

Let X be a distribution on $\mathbb{R}^p$ and assume $X = A\varepsilon$, where $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_q)$ is a vector of independent variables on $\mathbb{R}^q$ and $A \in \mathbb{R}^{p\times q}$ is a linear map, as in ICA (1). Then the d-th cumulant of X is the order d tensor

$$\kappa_d(X) = \sum_{i=1}^{q} \kappa_d(\varepsilon_i)\mathrm{a}_i^{\otimes d}, \tag{6}$$

where the scalar $\kappa_d(\varepsilon_i)$ is the d-th cumulant of variable ε_i and a_i is the i-th column of matrix A, as follows. The cumulants $\kappa_d(\varepsilon)$ are order d tensors of format $q \times \cdots \times q$. Since the variables ε_i are independent, by assumption, their cross-cumulants vanish [38, Section 2.1]. Hence the tensor $\kappa_d(\varepsilon)$ is diagonal: its entries vanish away from the $\kappa_d(\varepsilon_1), \ldots, \kappa_d(\varepsilon_q)$ on the main diagonal. A linear transformation of variables results in a multi-linear transformation of their cumulants. This gives the expression in (6), which writes the cumulant as a sum of symmetric rank one tensors.

If $q \leq p$ then $\kappa_d(X)$ has a unique rank q decomposition, whenever cumulants $\kappa_d(\varepsilon_i)$ are all non-zero and the columns of A are linearly independent, by [19]. Hence the vectors a_i can be recovered uniquely, up to permutation and scaling. This extends to $q > p$, as follows.

Proposition 2.1 *Assume that no pair of columns of $A \in \mathbb{R}^{p\times q}$ are collinear and that the q entries of ε are independent. Then, for d sufficiently large, all columns a_i with $\kappa_d(\varepsilon_i) \neq 0$ can be uniquely recovered, up to permutation and scaling, from the d-th cumulant of $X = A\varepsilon$.*

Proof For $m \geq q-1$, the tensors $\mathrm{a}_1^{\otimes m}, \ldots, \mathrm{a}_q^{\otimes m}$ are linearly independent, by [29, Proposition 4.3.7.6], since no pair of columns a_i are collinear. Let $d \geq 3m \geq 3(q-1)$ and consider $\kappa_d(X) = \sum_{i=1}^{q} \lambda_i \mathrm{a}_i^{\otimes d}$, where $\lambda_i := \kappa_d(\varepsilon_i)$. Consider its flattening of size $p^m \times p^m \times p^{d-2m}$. The decomposition of this flattened tensor is unique, by [19], since the vectors that appear in it are linearly independent. Hence the tensors $\mathrm{a}_i^{\otimes m}$ and $\mathrm{a}_i^{\otimes(d-2m)}$, and thus also the vectors a_i, can be uniquely recovered, up to permutation and scaling, for all indices i with $\lambda_i \neq 0$. □

For a sufficiently generic matrix A, one can recover the vectors uniquely, up to permutation and scaling, from the above tensor decomposition provided q is strictly less than the generic rank of an order d tensor of format $p \times \cdots \times p$, by [12]. The generic rank is $\lceil \frac{1}{p}\binom{p+d-1}{d} \rceil$ except for a finite list of pairs (p, q), see [29, Theorem

3.2.2.4]. Since for fixed p and large d, $\frac{1}{p}\begin{pmatrix} p+d-1 \\ d \end{pmatrix} \sim d^{p-1}$, this result allows for larger q relative to d than the condition $d \geq 3(q-1)$ coming from Proposition 2.1.

Corollary 2.2 *Assume that the entries of ε are independent and non-Gaussian and that no pair of columns of A are collinear. Then tensor decomposition of the cumulants of X recovers the matrix A, up to permutation and scaling of its columns.*

Proof The cumulant sequence $(\kappa_d(\varepsilon_i))_d$ has infinitely many non-zero terms, since ε_i is non-Gaussian [37]. Hence there are non-zero cumulants at high enough d to satisfy the hypotheses of Proposition 2.1. This is an alternative proof of [18, Theorems 1(i) and 3(i)]. □

The impossibility of recovering the columns without scaling ambiguity comes from the fact that we can extract or insert a global scalar from the factor $\mathrm{a}_i^{\otimes d}$. We have $(\lambda \mathrm{a})^{\otimes d} = \lambda^d \mathrm{a}^{\otimes d}$, hence

$$\kappa_d(X) = \begin{cases} \sum_{i=1}^q \left(\sqrt[d]{\kappa_d(\varepsilon_i)}\mathrm{a}_i \right)^{\otimes d} & d \text{ odd} \\ \sum_{i=1}^q \operatorname{sign}\left(\kappa_d(\varepsilon_i)\right) \left(\pm \sqrt[d]{|\kappa_d(\varepsilon_i)|}\mathrm{a}_i \right)^{\otimes d} & d \text{ even.} \end{cases} \tag{7}$$

Tensor decomposition will therefore recover the columns of A up to the factors $\pm \sqrt[d]{|\kappa_d(\varepsilon_i)|}$.

Consider the LCD setting of (3). We have $X = FZ = F(I-\Lambda)^{-1}\varepsilon$. The discussion above shows that the product $F(I-\Lambda)^{-1} \in \mathbb{R}^{p\times q}$ can be recovered (up to permutation and scaling), since the entries of the random vector ε are independent. However, it is not possible to recover the latent DAG $\mathcal{G}$ from the product $F(I-\Lambda)^{-1}$: a solution with empty DAG (that is, independent Z variables) is always consistent with the observations, since

$$F(I-\Lambda)^{-1}\varepsilon = \widetilde{F}\widetilde{Z},$$

where $\widetilde{F} = F(I-\Lambda)^{-1}$ and $\widetilde{Z} = \varepsilon$. This demonstrates the need for observations of X under multiple contexts.

2.2 Coupling Contexts

Distributions $X^{(k)}$ are linear mixtures of independent variables, since $X^{(k)} = F(I-\Lambda^{(k)})^{-1}\varepsilon^{(k)}$, where the entries of $\varepsilon^{(k)}$ are independent. Our goal is to recover the parameters F and $\Lambda^{(k)}$ for all $k \in K \cup \{0\}$. Our steps are as follows:

- using tensor decomposition, recover the products $F(I-\Lambda^{(k)})^{-1}$ for all $k \in K \cup \{0\}$, up to scaling and permutation of columns (Proposition 2.3);
- fix the scale and order of columns in the observational context $k = 0$, which recovers the matrix $F(I-\Lambda^{(0)})^{-1}$, using benign non-identifiability (Proposition

2.5, see Remark 1.4);
- find the permutation and scaling of columns for each $k \in K$ by comparing the columns of $F(I - \Lambda^{(0)})^{-1}$ to the matrix recovered from tensor decomposition of the *k*-th context (Corollary 2.10).

Then, in Sect. 3, we recover the parameters in F and $\Lambda^{(k)}$. We begin with the first step.

Proposition 2.3 *Consider LCD under Assumption 1.2. Then we can recover q and the matrices $F(I - \Lambda^{(k)})^{-1}$ up to scaling and permutation for all $k \in K \cup \{0\}$; i.e., we can recover*

$$F(I - \Lambda^{(k)})^{-1} D^{(k)} P^{(k)} \in \mathbb{R}^{p \times q}, \tag{8}$$

where $D^{(k)} \in \mathbb{R}^{q \times q}$ is diagonal, with non-zero diagonal entries, and $P^{(k)} \in \mathbb{R}^{q \times q}$ is a permutation matrix. The diagonal matrix $D^{(k)}$ can be assumed to have entries

$$D^{(k)}_{i,i} = \begin{cases} \sqrt[d_i]{\kappa_{d_i}(\varepsilon_i^{(k)})} & d_i odd, \\ \pm \sqrt[d_i]{|\kappa_{d_i}(\varepsilon_i^{(k)})|} & d_i even, \end{cases} \tag{9}$$

where, for all $i \in [q]$, d_i is large enough ($d_i \geq 3(q-1)$ suffices) and satisfies $\kappa_{d_i}(\varepsilon_i^{(k)}) \neq 0$.

Proof We have $X^{(k)} = A^{(k)} \varepsilon^{(k)}$, where $A^{(k)} = F(I - \Lambda^{(k)})^{-1}$ and k ranges over contexts $K \cup \{0\}$. We first prove the result under an additional assumption, that there exists a single number $d \geq 3$ that satisfies:

(a) the tensor decomposition

$$\kappa_d(X^{(k)}) = \sum_{i=1}^{q} \kappa_d(\varepsilon_i^{(k)}) (\mathrm{a}^{(k)})_i^{\otimes d} \tag{10}$$

is unique for all contexts k, where $(\mathrm{a}^{(k)})_i$ is the i-th column of $A^{(k)} = F(I - \Lambda^{(k)})^{-1}$,

(b) $\kappa_d(\varepsilon_i^{(k)}) \neq 0$ for all contexts k and all $i \in [q]$,

(c) $\kappa_d(\varepsilon_{i_k}^{(k)}) \neq \pm 1$ for all contexts k.

Fix such a d. No pair of columns of $A^{(k)}$ are collinear, since collinearity is a Zariski closed condition with non-empty complement and the entries of F and the non-zero values $\lambda_{i,j}^{(k)}$ are generic, by Assumption 1.2(b). Hence Proposition 2.1 applies, and we recover the mixing matrix $A^{(k)}$ up to permutation and scaling; i.e., we recover the matrices in (8). The number of columns of these matrices is q. Absorbing the coef-

ficients of the tensor decomposition into the vectors as in (7), the diagonal matrices in (8) satisfy (9) for every $i \in [q]$, $k \in K$, where $d_i = d$ for all i.

We now show why such a d as above is not required. Part (a) holds for any $d \geq 3(q-1)$, see the proof of Proposition 2.1. Part (b) is subtle: the existence of a sufficiently large d with $\kappa_d(\varepsilon_i^{(k)}) \neq 0$ is equivalent to Assumption 1.2(a) that the distribution $\varepsilon_i^{(k)}$ is non-Gaussian, by Marcinkiewicz's theorem [37]. However, this does not imply the existence of a *common* d with that property, as we assumed above. If such a common d does not exist, we instead recover the columns of $F(I-\Lambda^{(k)})^{-1}$ up to permutation and scaling, as well as the entries of $D^{(k)}$, using a *set* of large enough cumulants $\kappa_{d_1}(X^{(k)}), \ldots, \kappa_{d_m}(X^{(k)})$ such that for all i there exists $\ell \in [m]$ with $\kappa_{d_\ell}(\varepsilon_i^{(k)}) \neq 0$. The non-Gaussianity assures that such a set exists, and the number of non-collinear vectors recovered from these tensor decompositions is q. Part (c) can be avoided in the same way as (b), using Assumption 1.2(c). □

Column scaling and permutation as in Proposition 2.3 have natural interpretations in LCD: there is no natural order on the latent variables, and they can be re-scaled without affecting membership in the model, see Remark 1.4. The goal of this section is to show that it is possible to fix an order and scaling of latent variables that is consistent across contexts. The upshot is the following result.

Proposition 2.4 *Consider LCD under Assumption 1.2. Then we can recover the number of latent nodes q and the matrices*

$$A^{(k)} := F(I-\Lambda^{(k)})^{-1} \quad forall \quad k \in K \cup \{0\}. \tag{11}$$

We delay the the proof of this result and focus on some intermediate steps. We fix a scaling of errors and an order on latent variables when $k = 0$, as follows.

Proposition 2.5 *Without loss of generality $P^{(0)} = D^{(0)} = I$.*

Proof We have recovered ADP for some scaling D and permutation P, by Proposition 2.3, where we drop the superscripts since we refer only to the observational context. The permutation P orders the latent variables. We fix it to be the identity, thereby fixing an order of latent variables. We now consider D. Define $\widetilde{F} = FD$ and $\widetilde{\Lambda} = D^{-1}\Lambda D$. Then

$$F(I-\Lambda)^{-1}D = \widetilde{F}(I-\widetilde{\Lambda})^{-1}.$$

and matrices Λ and $\widetilde{\Lambda}$ have the same support. Hence $\widetilde{F}$ and $\widetilde{\Lambda}$ are valid parameters in the model, so we can without loss of generality set $D = I$. □

The choice in Proposition 2.5 sets a non-zero cumulant $\kappa_{d_i}(\varepsilon_i^{(0)})$ to ± 1 for each $i \in [q]$, see (9). Hence $D_{i,i}^{(k)} = \pm 1$ for all $i \neq i_k$, by Proposition 2.3, since $\varepsilon^{(k)}$ and $\varepsilon^{(0)}$ differ only at the intervention target i_k. We now compare $A^{(0)}$ and $A^{(k)}D^{(k)}P^{(k)}$. The parents of a node j are the set $\mathrm{pa}_{\mathcal{G}}(j) = \{i \in \mathcal{G} \mid i \to j \in \mathcal{G}\}$ and the ancestors

of j are $\mathrm{an}_{\mathcal{G}}(j) = \{i \in \mathcal{G} \mid i \to \cdots \to j \in \mathcal{G}\}$. We drop the subscript whenever $\mathcal{G}$ is fixed.

Remark 2.6 *(Paths in $\mathcal{G}$)* Entry (i, j) of the matrix $(I - \Lambda^{(k)})^{-1}$ is a sum over all the paths $j \to \cdots \to i$ in $\mathcal{G}$, where each path contributes the product $\lambda^{(k)}_{m,n}$ over all edges $n \to m$ in the path. For instance, for the DAG $3 \to 2 \to 1$ we have $(I - \Lambda^{(k)})^{-1}_{1,3} = \lambda^{(k)}_{1,2}\lambda^{(k)}_{2,3}$. Adding the edge $3 \to 1$ gives $(I - \Lambda^{(k)})^{-1}_{1,3} = \lambda^{(k)}_{1,2}\lambda^{(k)}_{2,3} + \lambda^{(k)}_{1,3}$. The entries of $F(I - \Lambda^{(k)})^{-1}$ extend these paths to the observed variables. See Fig. 1 and Example 3.4.

Proposition 2.7 *Recall that $i_k \in [q]$ is the intervention target of context k and let $j \in [q]$. Assume that F is generic and that the non-zero entries of $\Lambda^{(k)}$ are generic. Then one of three possibilities arises.*

(i) *$j = i_k$ and the j-th column of $A^{(0)}$ equals one of the columns of $A^{(k)}D^{(k)}P^{(k)}$ up to a scaling that is not ± 1;*
(ii) *$j \notin \mathrm{an}(i_k) \cup \{i_k\}$ and the j-th column of $A^{(0)}$ equals one of the columns of $A^{(k)}D^{(k)}P^{(k)}$, up to sign;*
(iii) *$j \in \mathrm{an}(i_k)$ and the j-th column of $A^{(0)}$ is not parallel to any of the columns of $A^{(k)}D^{(k)}P^{(k)}$.*

Proof We drop the factor of $P^{(k)}$ in the proof: it permutes the columns of $A^{(k)}D^{(k)}$ and we are reasoning only about the set of columns.

(i) Assume $j = i_k$. The (i, j) entry of $A^{(k)}D^{(k)}$ is

$$(A^{(k)}D^{(k)})_{i,j} = A^{(k)}_{i,j}D^{(k)}_{j,j} = A^{(0)}_{i,j}D^{(k)}_{j,j}, \tag{12}$$

where the second equality holds since $j = i_k$ is the intervention target and the entries of $A^{(k)}_{i,j}$ involve nodes that are non-ancestors of j (see Remark 2.6). Therefore, the j-th column of $A^{(k)}D^{(k)}$ is a non-trivial (not 0 or ± 1) multiple of the j-th column of $A^{(0)}$, since $D^{(k)}_{j,j} \neq \pm 1$ when j is the intervention target.

(ii) Assume $j \notin \mathrm{an}(i_k) \cup \{i_k\}$. The chain of equalities in (12) holds true, but $D^{(k)}_{j,j} = \pm 1$. Hence, the j-th column of $A^{(k)}D^{(k)}$ is the j-th column of $A^{(0)}$, up to sign.

(iii) Let $j \in \mathrm{an}(i_k)$. Assume for contradiction that there exists a column r of $A^{(k)}D^{(k)}$ that is parallel to the j-th column of $A^{(0)}$. Then there exists α such that for every $i \in [p]$,

$$\sum_{\ell \in [q]} f_{i,\ell}(I - \Lambda^{(0)})^{-1}_{\ell,j} = \alpha \sum_{\ell \in [q]} f_{i,\ell}(I - \Lambda^{(k)})^{-1}_{\ell,r}D^{(k)}_{r,r},$$

where $f_{i,\ell}$ is the (i,ℓ) entry of F. By genericity of F and $\Lambda^{(k)}$, the equality holds if and only if it holds for the coefficient of every $f_{i,\ell}$ independently. It is therefore equivalent to

$$(I-\Lambda^{(0)})^{-1}_{\ell,j} = \alpha (I-\Lambda^{(k)})^{-1}_{\ell,r} D^{(k)}_{r,r}$$

for all $\ell \in [q]$. If $\ell = j$, then by genericity of $\Lambda^{(k)}$ we have $r = j$ and $\frac{1}{\alpha} = D^{(k)}_{j,j}$. However, since $D^{(k)}_{j,j} = \pm 1$, this leads to the equality

$$(I-\Lambda^{(0)})^{-1}_{\ell,j} = (I-\Lambda^{(k)})^{-1}_{\ell,j}$$

for every $\ell \in [q]$, which implies by genericity that $\lambda^{(0)}_{i_k,m} = \lambda^{(k)}_{i_k,m}$ for every m, a contradiction. □

Proposition 2.9 recovers the target of each intervention. It also recovers the ancestors of each latent node. That is, it recovers the transitive closure $\overline{\mathcal{G}}$, providing a simpler proof of the following result, proven without the non-Gaussian assumption in [50, Theorem 1].

Corollary 2.8 *Consider LCD under Assumption 1.2 with one intervention (either perfect or soft) on each latent node. Then we can recover the transitive closure $\overline{\mathcal{G}}$ of the latent DAG $\mathcal{G}$.*

Proposition 2.7 partially recovers the permutation $P^{(k)}$, as it pairs all latent nodes $j \notin \text{an}(i_k)$ with the column of $F(I-\Lambda^{(0)})^{-1}$ indexed by j. We can therefore assume without loss of generality that $i_k = k$ and that $P^{(k)}_{i,j} = \delta_{i,j}$ for every $j \notin \text{an}(k)$. We are left to find the columns of $j \in \text{an}(k)$.

Proposition 2.9 *For $j_1, j_2 \in \text{an}(k)$, there exists $\alpha \in \mathbb{R}$ such that*

$$(I-\Lambda^{(0)})^{-1}_{i,j_1} - \left((I-\Lambda^{(k)})^{-1} D^{(k)}\right)_{i,j_2} = \alpha \left((I-\Lambda^{(0)})^{-1}_{i,k} - \left((I-\Lambda^{(k)})^{-1} D^{(k)}\right)_{i,k}\right)$$

for all $i \in [q]$, if and only if $j_1 = j_2$ and $D^{(k)}_{j_1,j_1} = 1$.

Proof Fix $j := j_1 = j_2$ and assume $D^{(k)}_{j_1,j_1} = 1$. The left hand side is a sum over all paths from node j to node i, through node k, since the paths that do not go through k cancel:

$$\begin{aligned}
(I-\Lambda^{(0)})^{-1}_{i,j} - (I-\Lambda^{(k)})^{-1}_{i,j} &= (I-\Lambda^{(0)})^{-1}_{i,k}(I-\Lambda^{(0)})^{-1}_{k,j} - (I-\Lambda^{(k)})^{-1}_{i,k}(I-\Lambda^{(k)})^{-1}_{k,j} \\
&= (I-\Lambda^{(0)})^{-1}_{i,k}\left((I-\Lambda^{(0)})^{-1}_{k,j} - (I-\Lambda^{(k)})^{-1}_{k,j}\right) \\
&= \tfrac{(I-\Lambda^{(0)})^{-1}_{k,j} - (I-\Lambda^{(k)})^{-1}_{k,j}}{1-D^{(k)}_{k,k}} \left((I-\Lambda^{(0)})^{-1}_{i,k} - (I-\Lambda^{(k)})^{-1}_{i,k} D^{(k)}_{k,k}\right),
\end{aligned}$$

where we used $(I-\Lambda^{(0)})^{-1}_{i,k} - (I-\Lambda^{(k)})^{-1}_{i,k} D^{(k)}_{k,k} = (I-\Lambda^{(0)})^{-1}_{i,k}(1-D^{(k)}_{k,k})$. This proves one direction.

Assume conversely that the equality in the statement holds for some $j_1, j_2 \in \mathrm{an}(k)$, and let $i = j_1$. Then

$$(I-\Lambda^{(0)})^{-1}_{j_1,j_1} - \left((I-\Lambda^{(k)})^{-1}D^{(k)}\right)_{j_1,j_2} = \alpha\left((I-\Lambda^{(0)})^{-1}_{j_1,k} - \left((I-\Lambda^{(k)})^{-1}D^{(k)}\right)_{j_1,k}\right).$$

The right-hand side is zero since the latent graph is a DAG and $j_1 \in \mathrm{an}(k)$. Hence

$$0 = (I-\Lambda^{(0)})^{-1}_{j_1,j_1} - \left((I-\Lambda^{(k)})^{-1}D^{(k)}\right)_{j_1,j_2} = 1 \pm (I-\Lambda^{(0)})^{-1}_{j_1,j_2}$$

where we used that there are no paths from j_2 to k and $D^{(k)}_{j_2,j_2} = \pm 1$. Therefore, $(I-\Lambda^{(0)})^{-1}_{j_1,j_2} = 1$, which implies by genericity that $j_1 = j_2$ and $D^{(k)}_{j_2,j_2} = 1$. □

Corollary 2.10 *For every k and for generic parameters in $F, \Lambda^{(0)}, \Lambda^{(k)}$, we have*

$$\mathrm{rank}\left(A^{(0)} - A^{(k)} D^{(k)} P^{(k)}\right) = 1$$

if and only if $P^{(k)} = I$ and $D^{(k)}_{j,j} = 1$ for all $j \neq k$.

Proof A matrix has rank one if and only if all its columns are scalar multiples. Therefore, our claim is equivalent to the existence for every $j \in [q]$ of some $\alpha \in \mathbb{R}$ such that

$$\begin{aligned}&\sum_{i\in[q]} f_{\ell,i}\left((I-\Lambda^{(0)})^{-1}_{i,j} - \left((I-\Lambda^{(k)})^{-1}D^{(k)}P^{(k)}\right)_{i,j}\right)\\ &= \alpha \sum_{i\in[q]} f_{\ell,i}\left((I-\Lambda^{(0)})^{-1}_{i,k} - \left((I-\Lambda^{(k)})^{-1}D^{(k)}P^{(k)}\right)_{i,k}\right),\end{aligned} \tag{13}$$

for every $\ell \in [p]$. If we treat the parameters in $F, \Lambda^{(0)}, \Lambda^{(k)}, D^{(k)}_{k,k}$ as indeterminates, the equation holds if and only if all the summands are equal. Analogously, this is the case if the $f_{\ell,i}$ parameters are generic.

For $j \notin \mathrm{an}(k)$, we have $P^{(k)}_{i,j} = \delta_{i,j}$. Assume for contradiction that $D^{(k)}_{j,j} = -1$. Then, for $i = k$, the left-hand side of (13) is 0 and the right-hand side is $\alpha(1 - D^{(k)}_{k,k})$, which forces $\alpha = 0$. However, for $i = j$, the left-hand side is 2, so $\alpha \neq 0$, a contradiction. This forces $D^{(k)}_{j,j} = 1$ for the non-ancestors of k. Putting this together with Proposition 2.9, we deduce that the matrix $(I-\Lambda^{(0)})^{-1} - (I-\Lambda^{(k)})^{-1}D^{(k)}P^{(k)}$ has rank at most 1 if and only if $P^{(k)} = I$ and $D^{(k)}_{j,j} = 1$ for all $j \neq k$. Moreover,

because $D^{(k)}_{k,k} \neq 1$, the k-th column of the difference matrix is non-zero, hence the rank is exactly 1. □

Proof of Proposition 2.4 We recover the matrices $A^{(k)}$ up to scaling and permutation, by Proposition 2.3. The upshot of Corollary 2.10 is that we can identify the target i_k of the intervention and the permutation. Hence we can get rid of $P^{(k)}$ by right multiplication with its transpose. Now the i_k-th column of $F(I - \Lambda^{(0)})^{-1}$ differs from the i_k-th column of $F(I - \Lambda^{(k)})^{-1}D^{(k)}$ by the scaling $D^{(k)}_{i_k,i_k}$, so we can also recover the diagonal matrix, and hence $A^{(k)}$ itself. □

Remark 2.11 While Proposition 2.4 holds for any $p, q \geq 2$, the proof is simpler when $q \leq p$. Then, the Moore-Penrose pseudo-inverse satisfies

$$\left(F(I - \Lambda^{(k)})^{-1}D^{(k)}P^{(k)}\right)^{+} = (P^{(k)})^{\top}(D^{(k)})^{-1}(I - \Lambda^{(k)})F^{+}.$$

Finding the permutation and intervention targets is done as follows. There is just one row in the pseudo-inverse of the context k that does not appear in the pseudo-inverse of the observational context. Hence it indexes the intervention target. The permutation is found by matching the remaining rows of the two matrices. The expression relating the psuedo-inverse of the product to the product of psuedo-inverses does not hold in general when $q > p$.

3 Recovery via Interventions

In this section, we identify when two latent graphs and parameters $F, \Lambda^{(k)}$ give the same distributions $X^{(k)}$. At this stage, we have access to the matrices $A^{(k)}$ in (11), by Proposition 2.4.

Proposition 3.1 *Distributions $F(I - \Lambda^{(k)})^{-1}\varepsilon^{(k)}$ and $\widetilde{F}(I - \widetilde{\Lambda}^{(k)})^{-1}\widetilde{\varepsilon}^{(k)}$ coincide for all $k \in K \cup \{0\}$ if and only if there exists a reordering of the sets $\{\varepsilon_i^{(0)}\}$, $\{\widetilde{\varepsilon}_i^{(0)}\}$ and a rescaling of $F, \Lambda^{(0)}, \widetilde{F}, \widetilde{\Lambda}^{(0)}$ via (5) such that $F(I - \Lambda^{(k)})^{-1} = \widetilde{F}(I - \widetilde{\Lambda}^{(k)})^{-1}$ for all $k \in K \cup \{0\}$.*

Proof Define $A^{(k)} = F(I - \Lambda^{(k)})^{-1}$ and $\widetilde{A}^{(k)} = \widetilde{F}(I - \widetilde{\Lambda}^{(k)})^{-1}$. The equality of matrices $A^{(k)}$ and $\widetilde{A}^{(k)}$ implies the equality of the distributions $X^{(k)} = A^{(k)}\varepsilon^{(k)}$ and $\widetilde{X}^{(k)} = \widetilde{A}^{(k)}\varepsilon^{(k)}$. Conversely, assume that distributions $X^{(k)}$ and $\widetilde{X}^{(k)}$ coincide. Then, we have the equality of cumulants $\kappa_d(X^{(k)}) = \kappa_d(\widetilde{X}^{(k)})$ for all k and d. To simplify the exposition, we assume that there exists d as in the proof of Proposition 2.3 (this assumption can be avoided using the same argument as in the proof of Proposition 2.3). For this fixed d and for each context k, the tensor decomposition of

$\kappa_d(X^{(k)})$ is unique up to rescaling and permutation. Since the cumulant is the same for both distributions, from the decomposition we get

$$F(I-\Lambda^{(k)})^{-1}D^{(k)}P^{(k)} = \widetilde{F}(I-\widetilde{\Lambda}^{(k)})^{-1}\widetilde{D}^{(k)}\widetilde{P}^{(k)}.$$

Fix $k=0$. We can reorder the variables $\varepsilon_i^{(0)}$ to set $P^{(0)}=I$ and we can absorb $D^{(0)}$ into F and $\Lambda^{(0)}$, as in Proposition 2.5. Analogously, we can do the same in the tilde setting. Therefore, up to reordering and rescaling via (5) we have

$$F(I-\Lambda^{(0)})^{-1} = \widetilde{F}(I-\widetilde{\Lambda}^{(0)})^{-1}.$$

Then, Corollary 2.10 implies that there exists a unique choice of signs of the diagonal matrices and a unique permutation matrix Q satisfying

$$\operatorname{rank}\left(A^{(0)} - A^{(k)}D^{(k)}P^{(k)}Q\right) = 1 = \operatorname{rank}\left(\widetilde{A}^{(0)} - \widetilde{A}^{(k)}\widetilde{D}^{(k)}\widetilde{P}^{(k)}Q\right),$$

and $Q=(P^{(k)})^\top$ and $Q=(\widetilde{P}^{(k)})^\top$. Therefore, $P^{(k)}=\widetilde{P}^{(k)}$, and by comparing the intervened columns of the difference matrices we have $D^{(k)}=\widetilde{D}^{(k)}$. This implies $A^{(k)}=\widetilde{A}^{(k)}$. □

The upshot is that solving Problem 1.3 is equivalent to solve the following problem.

Problem 3.2 Given a generic matrix $\widetilde{F}\in\mathbb{R}^{p\times q}$ and matrices $\widetilde{\Lambda}^{(0)},\dots,\widetilde{\Lambda}^{(q)}\in\mathbb{R}^{q\times q}$ constructed according to a model with DAG $\widetilde{\mathcal{G}}$, with generic non-zero entries, do there exist a generic matrix $F\in\mathbb{R}^{p\times q}$, and matrices $\Lambda^{(0)},\dots,\Lambda^{(q)}\in\mathbb{R}^{q\times q}$ constructed according to a model with DAG $\mathcal{G}$, such that

$$F(I-\Lambda^{(k)})^{-1} = \widetilde{F}(I-\widetilde{\Lambda}^{(k)})^{-1} \tag{14}$$

for all $k\in K\cup\{0\}$? If so, how are the DAGs and the corresponding matrices related? We solve the system of polynomial equations (14). The solution is unique if and only if the DAG and the matrices are identifiable. Otherwise, the set of solutions is the set of possible DAGs and space of possible parameters.

From now on, unless otherwise stated, we assume that we have the observational context and one intervention per latent node. We re-index contexts so that the k-th intervention (either soft or perfect) is on Z_k, hence $K=[q]$.

3.1 A Linear System

Let $A^{(k)}=\widetilde{F}(I-\widetilde{\Lambda}^{(k)})^{-1}$ and let $\mathcal{S}$ be the space of solutions to (14). The algebraic variety $\mathcal{S}$ is associated to the ideal

$$\mathcal{I} = \langle F(I-\Lambda^{(k)})^{-1} - A^{(k)},\ k=0,\dots,q\rangle.$$

The matrices $F,\Lambda^{(k)}$ are filled with indeterminates. Each point of $\mathcal{S}$ provides a graph and parameters compatible with the given model. At first sight, $\mathcal{S}$ might have high

degree, since the degree of the generators can reach $q+1$. However, there is a simpler set of generators:

$$\mathcal{I} = \langle F - A^{(k)}(I - \Lambda^{(k)}),\ k = 0, \ldots, q \rangle. \tag{15}$$

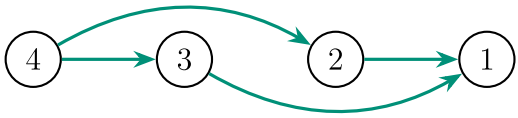
Assuming the $A^{(k)}$ are known, $\mathcal{I}$ has $(q+1)qp$ linear generators in a polynomial ring with $qp + (q+1)|e(\mathcal{G})|$ indeterminates a priori, namely the $f_{i,j}$, and all the non-zero entries of $\Lambda^{(k)}$ for $k = 0, \ldots, q$. To write down the equations for $\mathcal{I}$, we choose a candidate DAG $\mathcal{G}$. This can be the complete directed acyclic graph, or it could be sparser, if the model assumptions allow us to rule out some edges. We find a set of minimal generators for $\mathcal{I}$ to compute the dimension of the associated algebraic variety; i.e., to find the identifiability of the parameters (which will depend on our guess for $\mathcal{G}$).

Proposition 3.3 *Consider the setup in Assumption 1.2 with q soft interventions. When $d > 2$, the space of parameters F and $\Lambda^{(k)}$, $k \in [q] \cup \{0\}$, such that $\kappa_d(X^{(k)})$ is a given tensor is a linear space. When $d = 2$, for any $q \in \mathbb{N}$ there exists a DAG on q nodes such that the space of parameters for which $\kappa_2(X^{(k)})$ is a given matrix for all $k \in [q] \cup \{0\}$ is non-linear.*

Proof The case $d > 2$ follows from (15). For the case $d = 2$, consider a model on two latent nodes with one edge $2 \to 1$, with parameters $\widetilde{F} = \left(\begin{smallmatrix} 2 & 3 \\ 5 & 11 \end{smallmatrix}\right)$, $\widetilde{\lambda}^{(0)}_{1,2} = \widetilde{\lambda}^{(2)}_{1,2} = 7$, $\widetilde{\lambda}^{(1)}_{1,2} = 13$. Symbolic computation with, e.g., Macaulay2 or Oscar.jl shows that the space of parameters that satisfy $\kappa_2(X^{(k)}) = \kappa_2(\widetilde{X}^{(k)})$ for $k \in \{0,1,2\}$ is 1-dimensional and of degree 8. It is the union of 6 irreducible components, four linear and two quadratic. The same happens for generic parameters. We can embed this DAG into a DAG on q nodes with only one edge $2 \to 1$. Then, the space of solutions has the same dimension ($= 1$) and degree ($= 8$) as the space of solutions for the DAG on two nodes. □

Proposition 3.3 shows that the non-Gaussianity assumption is required in Theorem 1.7. We conclude this subsection with an example, to see the linear structure of $\mathcal{I}$.

Example 3.4 Consider the latent DAG

$\mathcal{G} =$ 

with parameters

$$F = \begin{pmatrix} 2 & 6 & 10 & 1 \\ 2 & 9 & -3 & 8 \\ -8 & 4 & 7 & 2 \\ -9 & 8 & 2 & -5 \end{pmatrix}, \qquad \Lambda^{(0)} = \Lambda^{(4)} = \begin{pmatrix} 0 & 9 & 3 & 0 \\ 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\Lambda^{(1)} = \begin{pmatrix} 0 & -5 & 8 & 0 \\ 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad \Lambda^{(2)} = \begin{pmatrix} 0 & 9 & 3 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad \Lambda^{(3)} = \begin{pmatrix} 0 & 9 & 3 & 0 \\ 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Then, assuming known $\mathcal{G}$, the ideal $\mathcal{I}$ (15) is minimally generated by 21 linear polynomials

$$\begin{gathered} f_{1,1} - 2, \quad f_{2,1} - 2, \quad f_{3,1} + 8, \quad f_{4,1} + 9, \\ f_{1,2} + 2\lambda_{1,2}^{(0)} - 24, \quad f_{2,2} + 2\lambda_{1,2}^{(0)} - 27, \quad f_{3,2} - 8\lambda_{1,2}^{(0)} + 68, \quad f_{4,2} - 9\lambda_{1,2}^{(0)} + 73, \\ f_{1,3} + 2\lambda_{1,3}^{(0)} - 16, \quad f_{2,3} + 2\lambda_{1,3}^{(0)} - 3, \quad f_{3,3} - 8\lambda_{1,3}^{(0)} + 17, \quad f_{4,3} - 9\lambda_{1,3}^{(0)} + 25, \\ f_{1,4} + \tfrac{172}{7}\lambda_{3,4}^{(0)} - 173, \quad f_{2,4} + \tfrac{177}{14}\lambda_{3,4}^{(0)} - \tfrac{193}{2}, \quad f_{3,4} - \tfrac{289}{7}\lambda_{3,4}^{(0)} + 287, \quad f_{4,4} - \tfrac{715}{14}\lambda_{3,4}^{(0)} + \tfrac{725}{2}, \\ \lambda_{1,2}^{(1)} - \lambda_{1,2}^{(0)} + 14, \quad \lambda_{1,3}^{(1)} - \lambda_{1,3}^{(0)} - 5, \quad \lambda_{2,4}^{(2)} - \lambda_{2,4}^{(0)} + 8, \quad \lambda_{3,4}^{(3)} - \lambda_{3,4}^{(0)} + 8, \quad -14\lambda_{2,4}^{(0)} + 5\lambda_{3,4}^{(0)} + 105. \end{gathered}$$

These can be found by computing the primary decomposition of $\mathcal{I}$ in computer algebra software, such as Macaulay2 [36] or Oscar.jl [14, 41].

3.2 Perfect Interventions

When the interventions are perfect, namely $\lambda_{k,j}^{(k)} = 0$ for every $k \in [q]$, there is a unique solution to the linear system in (15), provided the candidate DAG contains all edges of the true graph. In other words, the ideal $\mathcal{I}$ is zero dimensional and defines a point. This is Theorem 1.6.

Proof of Theorem 1.6 Worst case necessity of one intervention per node for identifiability is a direct consequence of [50, Proposition 5]. We prove sufficiency. We have matrices $A^{(k)} = F(I - \Lambda^{(k)})^{-1}$, by Proposition 2.4. Pick $k, j \in [q]$ with $k \neq j$. Then,

$$\begin{aligned} \left(A^{(0)} - A^{(k)}\right)_{1,j} &= \sum_{\ell \in [q]} f_{1,\ell} \left((I - \Lambda^{(0)})^{-1} - (I - \Lambda^{(k)})^{-1}\right)_{\ell,i} \\ &= \sum_{\ell \in \operatorname{de}(k)} f_{1,\ell} (I - \Lambda^{(0)})^{-1}_{\ell,k} \left((I - \Lambda^{(0)})^{-1} - (I - \Lambda^{(k)})^{-1}\right)_{k,j} \\ &= A^{(0)}_{1,k} \, (I - \Lambda^{(0)})^{-1}_{k,j}. \end{aligned}$$

With this, we construct $(I - \Lambda^{(0)})^{-1}$ and hence recover $\Lambda^{(0)}$. We multiply $A^{(0)}(I - \Lambda^{(0)})$ to obtain F. □

The above result shows that q perfect interventions are sufficient to recover the DAG and the parameters of a model. To find the parameters (and hence the latent DAG), one can solve the linear system (15) or follow the procedure in the proof.

Remark 3.5 When $q \leq p$, an alternative proof via pseudo-inverses exists, see Sect. 4.3.

When $q > p$, the non-Gaussianity assumption is necessary for Theorem 1.6, as follows.

Proposition 3.6 *Consider LCD under Assumption 1.2 with perfect interventions and $q > p$. Then one perfect intervention on each latent node is not sufficient to recover the latent DAG $\mathcal{G}$ and the parameters F and $\Lambda^{(k)}$ from the covariance matrices of $X^{(k)}$.*

Proof For $(p, q) = (2, 3)$ and $\mathcal{G} = \emptyset$, we compute the parameters F and $\Lambda^{(k)}$ for which the covariance matrices $F(I - \Lambda^{(k)})^{-1}(D^{(k)})^2(I - \Lambda^{(k)})^{-\top}F^\top$ coincide with the true covariance matrices for $k = 0, 1, 2$. We choose and fix an ordering of the nodes, and we fix the scaling by imposing $D^{(0)} = I$. This space has dimension 2, so the parameters cannot be recovered uniquely. We can embed this DAG in a DAG with q nodes, for any q. Hence the $p \times p$ covariance matrices do not contain enough information to recover the parameters, when $p < q$. □

3.3 Soft Interventions

In this section we compute the dimension of solutions of the linear system $F - A^{(k)}(I - \Lambda^{(k)}) = 0$ for $k = 0, \ldots, q$, under soft interventions. For every k and for every $\ell \in [p], j \in [q]$, we have

$$f_{\ell,j} + \sum_{i \in \mathrm{ch}(j)} A^{(k)}_{\ell,i}\lambda^{(k)}_{i,j} = A^{(k)}_{\ell,j}.$$

Since we consider single-node soft interventions, there are $pq(q+1)$ equations in $pq + 2|e(\mathcal{G})|$ indeterminates, namely $f_{\ell,j}$ for all $\ell \in [p]$, $j \in [q]$, and $\lambda^{(0)}_{i,j}$ for all $(j \to i) \in e(\mathcal{G})$, and $\lambda^{(k)}_{k,j}$ for all $(j \to k) \in e(\mathcal{G})$. For each (ℓ, j), we subtract the equation for $k = 0$ from the equations for $k \in [q]$. Then the $\big(pq(q+1)\big) \times \big(pq + 2|e(\mathcal{G})|\big)$ matrix of the linear system has block structure

$$\left(\begin{array}{c|c} I_{pq} & \star \\ \hline 0 & \star \end{array}\right).$$

We can focus on the $\big(pq^2\big) \times \big(2|e(\mathcal{G})|\big)$ bottom-right block, involving only the indeterminates $\Lambda^{(k)}$. The equations of this smaller linear system are $A^{(k)}(I - \Lambda^{(k)}) - A^{(0)}(I - \Lambda^{(0)}) = 0$, or

$$\sum_{i \in \mathrm{ch}(j)} A^{(k)}_{\ell,i}\lambda^{(k)}_{i,j} - \sum_{i \in \mathrm{ch}(j)} A^{(0)}_{\ell,i}\lambda^{(0)}_{i,j} = A^{(k)}_{\ell,j} - A^{(0)}_{\ell,j}, \tag{16}$$

for $k, j \in [q]$, $\ell \in [p]$. There are three cases:

(1.) If $k \notin \mathrm{ch}(j)$, then (16) becomes

$$\sum_{i\in \mathrm{ch}(j)} (A^{(k)}_{\ell,i} - A^{(0)}_{\ell,i})\lambda^{(0)}_{i,j} = (A^{(k)}_{\ell,j} - A^{(0)}_{\ell,j}). \tag{17}$$

The (ℓ, i) entry of $A^{(k)} - A^{(0)}$ is by definition $\sum_{n\in \mathrm{de}(i)} f_{\ell,n}\left((I-\Lambda^{(k)})^{-1} - (I-\Lambda^{(0)})^{-1}\right)_{n,i}$. If $k \notin \mathrm{de}(j)$, then $\left((I-\Lambda^{(k)})^{-1} - (I-\Lambda^{(0)})^{-1}\right)_{n,i} = 0$ for every n since by construction $\mathrm{de}(j) \supset \mathrm{de}(i)$. Hence, (17) reads $0 = 0$ and it imposes no condition on our indeterminates.

(2.) If $k \in \mathrm{de}(j) \setminus \mathrm{ch}(j)$, then $\left((I-\Lambda^{(k)})^{-1} - (I-\Lambda^{(0)})^{-1}\right)_{n,i} \neq 0$, since there is a path from i to n through k. Such a path must exist for some n, since k is a descendant of some i. Hence the coefficients of (17) are non-zero, and we get linear conditions on the indeterminates.

(3.) Finally, if $k \in \mathrm{ch}(j)$, we get an expression for $\lambda^{(k)}_{k,j}$ in terms of the $\lambda^{(0)}_{i,j}$:

$$\begin{aligned}\lambda^{(k)}_{k,j} &= \frac{1}{A^{(k)}_{\ell,k}}\left(A^{(0)}_{\ell,k}\lambda^{(0)}_{k,j} + \sum_{\substack{i\in \mathrm{ch}(j)\\ i\neq k}} (A^{(0)}_{\ell,i} - A^{(k)}_{\ell,i})\lambda^{(0)}_{i,j} + (A^{(k)}_{\ell,j} - A^{(0)}_{\ell,j})\right) \\ &= \lambda^{(0)}_{k,j} + \sum_{\substack{i\in \mathrm{ch}(j)\\ i\neq k}} \frac{A^{(0)}_{\ell,i} - A^{(k)}_{\ell,i}}{A^{(0)}_{\ell,k}}\lambda^{(0)}_{i,j} + \frac{A^{(k)}_{\ell,j} - A^{(0)}_{\ell,j}}{A^{(0)}_{\ell,k}},\end{aligned} \tag{18}$$

where we used $A^{(k)}_{\ell,k} - A^{(0)}_{\ell,k} = 0$ because $\left((I-\Lambda^{(k)})^{-1} - (I-\Lambda^{(0)})^{-1}\right)_{n,k} = 0$ for every n. We get (18) for every $\ell \in [p]$. However, most equations are redundant.

The following result mimics Proposition 2.9.

Proposition 3.7 *For $k \in [q]$, let $\Delta^{(k)} = (I-\Lambda^{(k)})^{-1} - (I-\Lambda^{(0)})^{-1}$. Then, $\mathrm{rank}\left(\Delta^{(k)}\right) \leq 1$, with equality if and only if $\mathrm{an}(k) \neq \emptyset$.*

Proof Fix $k \in [q]$ and recall that the (i, j) entry of $(I-\Lambda^{(k)})^{-1}$ is the sum of all paths from Z_j to Z_i, where a path is encoded as the product of $\lambda^{(k)}_{m,n}$ for all edges $n \to m$ in the path. Then, the only non-zero columns of $\Delta^{(k)}$ are those indexed by j for $j \in \mathrm{an}(k)$. We prove that these columns are multiple of each other. Let $j_1, j_2 \in \mathrm{an}(k)$, then

$$\begin{aligned}\Delta^{(k)}_{i,j_m} &= (I-\Lambda^{(k)})^{-1}_{i,k}(I-\Lambda^{(k)})^{-1}_{k,j_m} - (I-\Lambda^{(0)})^{-1}_{i,k}(I-\Lambda^{(0)})^{-1}_{k,j_m} \\ &= (I-\Lambda^{(0)})^{-1}_{i,k}\Delta^{(k)}_{k,j_m}\end{aligned}$$

for $m = 1, 2$. Hence, for every $i \in [q]$, the (i, j_1) entry equals the (i, j_2) entry up to $\frac{\Delta^{(k)}_{k,j_2}}{\Delta^{(k)}_{k,j_1}}$. □

For generic parameters we have $\operatorname{rank}(A^{(k)} - A^{(0)}) \leq 1$ for all $k \in [q]$, with equality whenever $\operatorname{an}(k) \neq \emptyset$, by Proposition 3.7, with proof is analogous to that of Corollary 2.10. Hence the conditions in (17) are equivalent for all $\ell \in [p]$, and the same is true of the conditions in (18). This reduces the size of the linear system, taking only the equations for $\ell = 1 \in [p]$. We obtain a reduced matrix of the linear system

$$\left(\begin{array}{c|c|c} I_{pq} & 0 & \star \\ \hline 0 & I_{|e(\mathcal{G})|} & \star \\ \hline 0 & 0 & \star \end{array}\right), \tag{19}$$

where the top block writes F in terms of $\Lambda^{(0)}$, the second block writes $\Lambda^{(k)}$ in terms of $\Lambda^{(0)}$, and the bottom block gives the conditions (17) on $\Lambda^{(0)}$. The latter are $\sum_{j\in[q]} |\operatorname{de}(j) \setminus \operatorname{ch}(j)|$ equations in $\sum_{j\in[q]} |\operatorname{ch}(j)| = |e(\mathcal{G})|$ indeterminates. The conditions are independent for each j. Namely, the block has the form

$$M = \left(\begin{array}{c|c|c|c} M[1] & 0 & \cdots & 0 \\ \hline 0 & M[2] & \cdots & 0 \\ \hline \vdots & \cdots & \ddots & \vdots \\ \hline 0 & \cdots & 0 & M[q] \end{array}\right).$$

Each sub-block has size $|\operatorname{de}(j) \setminus \operatorname{ch}(j)| \times |\operatorname{ch}(j)|$ and defines $M[j] \cdot \left(\lambda^{(0)}_{i,j}\right)_{i\in \operatorname{ch}(j)} = b[j]$ where

$$M[j] = \left(\left(A^{(k)} - A^{(0)}\right)_{1,i}\right),\ k \in \operatorname{de}(j) \setminus \operatorname{ch}(j),\ i \in \operatorname{ch}(j),$$
$$b[j] = \left(\left(A^{(k)} - A^{(0)}\right)_{1,j}\right),\ k \in \operatorname{de}(j) \setminus \operatorname{ch}(j).$$

At this point, it seems that the matrices defining the linear system depend on F and $\Lambda^{(k)}$. However, following the proof of Proposition 3.7, we have

$$\left(A^{(k)} - A^{(0)}\right)_{1,i} = \sum_{n\in[q]} f_{1,n}\, \Delta^{(k)}_{n,i} = \sum_{n\in \operatorname{de}(k)} f_{1,n} (I - \Lambda^{(0)})^{-1}_{n,k}\, \Delta^{(k)}_{k,i} = A^{(0)}_{1,k}\, \Delta^{(k)}_{k,i}.$$

Assuming $A^{(0)}_{1,k} \neq 0$ for every $k \in [q]$, which holds generically, we can rescale to obtain

$$M[j] = \left(\ \Delta^{(k)}_{k,i}\ \right),\ k \in \operatorname{de}(j) \setminus \operatorname{ch}(j),\ i \in \operatorname{ch}(j),$$
$$b[j] = \left(\ \Delta^{(k)}_{k,j}\ \right),\ k \in \operatorname{de}(j) \setminus \operatorname{ch}(j), \tag{20}$$

where $\Delta^{(k)} = (I - \Lambda^{(k)})^{-1} - (I - \Lambda^{(0)})^{-1}$. From this, we see that $M[j]$ and $b[j]$ depend only on the latent DAG and its parameters: the linear system (17) becomes

$$\sum_{i \in \mathrm{ch}(j)} \Delta^{(k)}_{k,i} \lambda^{(0)}_{i,j} = \Delta^{(k)}_{k,j}, \tag{21}$$

for all $j \in [q]$. We compute the dimension of the solution space by comparing the ranks $|\,\mathrm{de}(j) \setminus \mathrm{ch}(j)| \times |\,\mathrm{ch}(j)|$ matrix $M[j]$ and the $|\,\mathrm{de}(j) \setminus \mathrm{ch}(j)| \times (|\,\mathrm{ch}(j)| + 1)$ matrix $(M[j]|b[j])$. Recall that an ideal has dimension -1 when its associated variety is empty.

Proposition 3.8 *Assume that the interventions are soft. For each node j, let*

$$c_j = \begin{cases} -1 & \text{if } \operatorname{rank} M[j] \neq \operatorname{rank} (M[j]|b[j]), \\ |\,\mathrm{ch}(j)| - \operatorname{rank} M[j] & \text{otherwise,} \end{cases}$$

where M[j] and b[j] are defined in (20). Then, the ideal $\mathcal{I}$ in (15) has dimension

$$\dim \mathcal{I} = \begin{cases} -1 & \text{if } c_j = -1 \text{ for some } j \in [q], \\ \sum_{j=1}^{q} c_j & \text{otherwise.} \end{cases}$$

Proof The dimension of $\mathcal{I}$ is the dimension of the solution space of (21), that is,

$$M[j] \cdot \left(\lambda^{(0)}_{i,j}\right)_{i \in \mathrm{ch}(j)} = b[j]$$

for all $j \in [q]$, by (19). Its dimension c_j is $|\,\mathrm{ch}(j)| - \operatorname{rank} M[j]$ if $\operatorname{rank} M[j] \neq \operatorname{rank} (M[j]|b[j])$. Otherwise, the solution space is empty and we set $c_j = -1$, as is convention. □

Corollary 3.9 *With one soft intervention per latent node it is never possible to recover uniquely all the parameters of the model.*

Proof The result remains true if we assume knowledge of the latent DAG $\mathcal{G}$. Let $\mathrm{ch}(i) = \emptyset$. Take $j \in \mathrm{pa}(i)$ such that $\mathrm{de}(j) \setminus \mathrm{ch}(j) = \emptyset$. Then $M[j] = \emptyset$, so $c_j = |\,\mathrm{ch}(j)| \geq 1$. Therefore, $\dim \mathcal{I} \geq 1$ and it is not possible to identify uniquely the parameters $f_{\ell,j}$ and $\lambda^{(k)}_{k,j}$, for $\ell \in [p]$ and $k \in \mathrm{ch}(j) \setminus \{i\}$. □

Adding interventions does not affect the matrices $M[j]$ in the proof of Corollary 3.9. Therefore Corollary 1.8 follows: non-identifiability holds regardless of the number of interventions.

When $c_j = 0$ it is possible to identify uniquely all parameters $\lambda^{(0)}_{i,j}$ for $i \in \mathrm{ch}(j)$, as well as $f_{\ell,j}$ and $\lambda^{(k)}_{k,j}$, for $\ell \in [p]$ and $k \in \mathrm{ch}(j)$. The condition $c_j = 0$ holds, for example, when $\mathrm{ch}(j) = \{i_1\}$ and $\mathrm{de}(j) = \{i_1, i_2\}$.

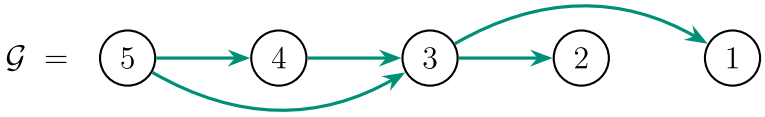
Example 3.10 We continue Example 3.4. The matrices are

$$M[1] = M[2] = M[3] = b[1] = b[2] = b[3] = \emptyset, \quad M[4] = (-14 \quad 5)\,, \; b[4] = -105,$$

hence $c_1 = |\,\mathrm{ch}(1)| = 0$, $c_2 = |\,\mathrm{ch}(2)| = 1$, $c_3 = |\,\mathrm{ch}(3)| = 1$, $c_4 = |\,\mathrm{ch}(4)| - \operatorname{rank} M[4] = 2 - 1 = 1$. By Proposition 3.8, we have $\dim \mathcal{I} = 3$ and in fact it is minimally generated by the 21 linear polynomials in 24 indeterminates in Example 3.4.

The rank of $M[j]$ depends on the structure of the DAG $\mathcal{G}$ beyond the number of children and descendants of j. This is highlighted in the following example.

Example 3.11 Consider the DAG



Node $j = 5$ has 2 children and 4 descendants, hence (21) consists of equations

$$\sum_{i=3,4} \Delta^{(k)}_{k,i} \lambda^{(0)}_{i,5} = \Delta^{(k)}_{k,5} \qquad k = 1, 2. \tag{22}$$

They impose conditions on the $\lambda^{(0)}_{i,j}$. There are two $\lambda^{(0)}_{i,j}$ from node 5, namely $\lambda^{(0)}_{3,5}, \lambda^{(0)}_{4,5}$, and two equations. However, the equations in (22) are dependent. We have

$$M[5] = \begin{pmatrix} \lambda^{(1)}_{1,3} - \lambda^{(0)}_{1,3} & (\lambda^{(1)}_{1,3} - \lambda^{(0)}_{1,3})\lambda^{(0)}_{3,4} \\ \lambda^{(2)}_{2,3} - \lambda^{(0)}_{2,3} & (\lambda^{(2)}_{2,3} - \lambda^{(0)}_{2,3})\lambda^{(0)}_{3,4} \end{pmatrix}, \qquad b[5] = \begin{pmatrix} (\lambda^{(1)}_{1,3} - \lambda^{(0)}_{1,3})(\lambda^{(0)}_{3,4}\lambda^{(0)}_{4,5} + \lambda^{(0)}_{3,5}) \\ (\lambda^{(2)}_{2,3} - \lambda^{(0)}_{2,3})(\lambda^{(0)}_{3,4}\lambda^{(0)}_{4,5} + \lambda^{(0)}_{3,5}) \end{pmatrix},$$

so $\operatorname{rank} M[5] = \operatorname{rank}\,(M[5]|b[5]) = 1 < 2$. Hence we cannot recover the parameters $\lambda^{(0)}_{3,5}, \lambda^{(0)}_{4,5}$. The reason $\operatorname{rank} M[5] < 2$ is that all the paths from 4 to 1 or 2 (encoded in the second column of $M[5]$) and all the paths from 5 to 1 or 2 (encoded in $b[5]$) go through 3. This factorization of paths creates dependencies in $M[5]$, $b[5]$, preventing identifiability.

To recover as many parameters as possible, DAGs should balance between too many children, hence too many indeterminates, and too few children, hence paths factorize more easily.

From an algorithmic point of view, we can check the rank condition in Proposition 3.8. Indeed, we have matrices $A^{(k)}$, and we can compute, for all $i, k \in [q]$ with $i \neq k$, the entries

$$\Delta^{(k)}_{k,i} = \frac{A^{(k)}_{1,i} - A^{(0)}_{1,i}}{A^{(0)}_{1,k}}.$$

Remark 3.12 If $q \leq p$, the computations can be simplified. We can write (16) as

$$\Lambda^{(k)} = (A^{(k)})^+ A^{(0)} \Lambda^{(0)} + I - (A^{(k)})^+ A^{(0)}.$$

This writes the $\lambda_{k,j}^{(k)}$ indeterminates in terms of $\lambda_{i,j}^{(0)}$ indeterminates, and enables us to find the linear conditions on the $\lambda_{i,j}^{(0)}$ indeterminates. The reduction of the linear system is the same as in (20), as can be proved by noticing that for any $k \neq i$, the k-th row of $(A^{(k)})^+ A^{(0)}$ equals the k-th row of $\Delta^{(k)}$ up to sign. Indeed,

$$\begin{aligned}\left((A^{(k)})^+ A^{(0)}\right)_{k,i} &= \left((I - \Lambda^{(k)})(I - \Lambda^{(0)})^{-1}\right)_{k,i} \\ &= \sum_{\ell \in \mathrm{pa}(k)} -\lambda_{k,\ell}^{(k)} (I - \Lambda_{\ell,i}^{(0)})^{-1} + (I - \Lambda^{(0)})_{k,i}^{-1} \\ &= -(I - \Lambda^{(k)})_{k,i}^{-1} + (I - \Lambda^{(0)})_{k,i}^{-1} = -\Delta_{k,i}^{(k)},\end{aligned}$$

where the last row used $(I - \Lambda_{\ell,i}^{(0)})^{-1} = (I - \Lambda_{\ell,i}^{(k)})^{-1}$ for every $\ell \in \mathrm{pa}(k)$.

3.3.1 Identifiability of the Latent DAG

For perfect interventions, Theorem 1.6 shows that we can recover the latent DAG of the model, as well as the parameters. With soft interventions, we cannot recover the whole DAG and all the parameters (see Corollary 3.9). It is natural to wonder to what extent we can recover the latent DAG. Thanks to Proposition 3.8, we can turn this into $2q$ rank computations.

Definition 3.13 Given a model with matrices $A^{(k)} \in \mathbb{R}^{p \times q}$ for $k = 0, \ldots, q$, we say that a DAG $\mathcal{G}'$ is *compatible* with the model if there exist parameters $F \in \mathbb{R}^{p \times q}$, $\Lambda^{(k)} \in \mathbb{R}^{q \times q}$ defined according to the latent DAG $\mathcal{G}'$, such that $A^{(k)} = F(I - \Lambda^{(k)})^{-1}$ for all k.

If the true latent DAG is $\mathcal{G}$, a compatible DAG $\mathcal{G}'$ must satisfy $\overline{\mathcal{G}} = \overline{\mathcal{G}'}$, by Corollary 2.8. To emphasize that a matrix such as $\Lambda^{(k)}$ depends on a DAG $\mathcal{G}$, we write $\Lambda_{\mathcal{G}}^{(k)}$. In the same spirit, we define

$$\begin{aligned}M_{\mathcal{G},\mathcal{G}'}[j] &= \left((\Delta_{\mathcal{G}}^{(k)})_{k,i} \right) \; k \in \mathrm{de}_{\mathcal{G}'}(j) \setminus \mathrm{ch}_{\mathcal{G}'}(j), \; i \in \mathrm{ch}_{\mathcal{G}'}(j), \\ b_{\mathcal{G},\mathcal{G}'}[j] &= \left((\Delta_{\mathcal{G}}^{(k)})_{k,j} \right) \; k \in \mathrm{de}_{\mathcal{G}'}(j) \setminus \mathrm{ch}_{\mathcal{G}'}(j),\end{aligned}$$

where the indexing depends on $\mathcal{G}'$ and $\Delta_{\mathcal{G}}^{(k)} = (I - \Lambda_{\mathcal{G}}^{(k)})^{-1} - (I - \Lambda_{\mathcal{G}}^{(0)})^{-1}$. Recall that

$$[(I - \Lambda^{(0)})^{-1}]_{\mathcal{R},\mathcal{C}}$$

denotes the submatrix with rows in $\mathcal{R} \subset [q]$ and columns in $\mathcal{C} \subset [q]$. We give the following definition, already mentioned in the introduction.

Definition 3.14 Given a DAG $\mathcal{G}$ on q nodes, its *soft-compatible* class is

$$\begin{aligned}\mathrm{soft}(\mathcal{G}) = \Big\{ & \mathcal{G}' \text{ DAG} \mid \overline{\mathcal{G}'} = \overline{\mathcal{G}} \text{ and for all } j \in [q] \\ & \mathrm{rank}[(I - \Lambda_{\mathcal{G}}^{(0)})^{-1}]_{\mathrm{de}_{\mathcal{G}'}(j)\setminus \mathrm{ch}_{\mathcal{G}'}(j),\ \mathrm{ch}_{\mathcal{G}'}(j)} = \mathrm{rank}[(I - \Lambda_{\mathcal{G}}^{(0)})^{-1}]_{\mathrm{de}_{\mathcal{G}'}(j)\setminus \mathrm{ch}_{\mathcal{G}'}(j),\ \overline{\mathrm{ch}}_{\mathcal{G}'}(j)} \Big\},\end{aligned} \tag{23}$$

where $\overline{\mathrm{ch}}(j) := \mathrm{ch}(j) \cup \{j\}$ for $j \in [q]$.
The soft-compatible class of a DAG $\mathcal{G}$ is the set of all graphs that are compatible with a model with latent DAG $\mathcal{G}$, as follows.

Theorem 3.15 *Consider LCD under under Assumption 1.2 with DAG $\mathcal{G}$. Then a graph $\mathcal{G}'$ is compatible with the model if and only if $\mathcal{G}' \in \mathrm{soft}(\mathcal{G})$.*

Proof The ranks of the matrices in (23) are the same as the ranks of the matrices $M_{\mathcal{G},\mathcal{G}'}[j]$ and $(M_{\mathcal{G},\mathcal{G}'}[j]|b_{\mathcal{G},\mathcal{G}'}[j])$, since the first matrices can be obtained from the second by replacing each term $\lambda_{k,i}^{(k)} - \lambda_{k,i}^{(0)}$ with $\lambda_{k,i}^{(0)}$. By the genericity of the parameters, this does not affect the rank. The genericity assumption allows us to switch between the parameters of the model and abstract indeterminates without change. Therefore, the ranks of the matrices in (23) coincide with the ranks of $M[j]$ and $(M[j]|b[j])$ from (20).

The DAG $\mathcal{G}'$ is compatible with the model if and only if the corresponding ideal $\mathcal{I}$ has non-negative dimension. Here the indeterminates of the system defined by $\mathcal{I}$ are the entries of F and of $\Lambda_{\mathcal{G}'}^{(k)}$. By Proposition 3.8, it is equivalent to $c_j \neq -1$ for all j, which holds if and only if $\mathrm{rank} M[j] = \mathrm{rank}\,(M[j]|b[j])$ for all j. This is equivalent to the condition $\mathcal{G}' \in \mathrm{soft}(\mathcal{G})$. □

Proof of Theorem 1.7 The linearity follows from equation (15). The positive dimensionality follows from Corollary 3.9. The compatibility class of DAGs is Theorem 3.15. □

We investigate the concept of soft-compatible class. If $\mathcal{G}'$ has the same transitive closure as $\mathcal{G}$ and if $\mathrm{ch}_{\mathcal{G}'}(j) \supset \mathrm{ch}_{\mathcal{G}}(j)$ for all j, then $\mathcal{G}' \in \mathrm{soft}(\mathcal{G})$. This is true because

$$[(I - \Lambda_{\mathcal{G}}^{(0)})^{-1}]_{\mathrm{de}_{\mathcal{G}}(j)\setminus \mathrm{ch}_{\mathcal{G}}(j),\ \mathrm{ch}_{\mathcal{G}}(j)} \quad \text{and} \quad [(I - \Lambda_{\mathcal{G}}^{(0)})^{-1}]_{\mathrm{de}_{\mathcal{G}}(j)\setminus \mathrm{ch}_{\mathcal{G}}(j),\ \overline{\mathrm{ch}}_{\mathcal{G}}(j)}$$

always satisfies the rank condition (since by construction a solution with DAG $\mathcal{G}$ exists). However, the matrix $[(I - \Lambda_{\mathcal{G}}^{(0)})^{-1}]_{\mathrm{de}_{\mathcal{G}'}(j)\setminus \mathrm{ch}_{\mathcal{G}'}(j),\ \mathrm{ch}_{\mathcal{G}'}(j)}$ is obtained from $[(I - \Lambda_{\mathcal{G}}^{(0)})^{-1}]_{\mathrm{de}_{\mathcal{G}}(j)\setminus \mathrm{ch}_{\mathcal{G}}(j),\ \mathrm{ch}_{\mathcal{G}}(j)}$ by adding columns and deleting rows, hence the column indexed by j remains in the span of the columns indexed by its children. Therefore, in order to exit the soft-compatible class of $\mathcal{G}$, a DAG $\mathcal{G}'$ with the same transitive closure must have enough more children at some node.

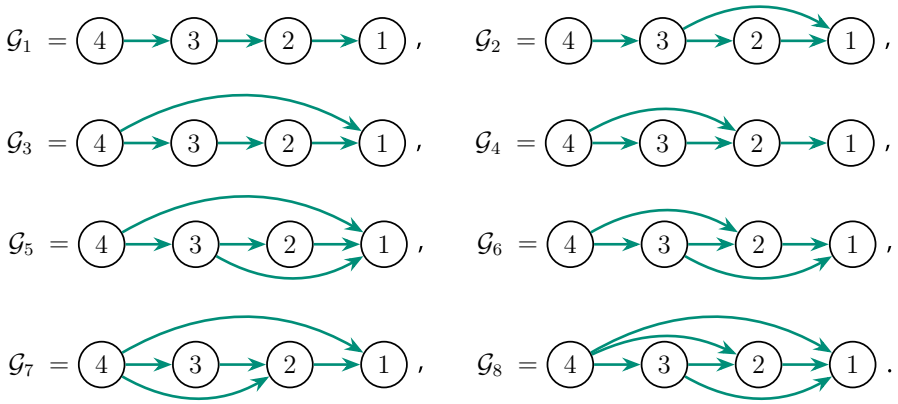
If $q = 2$, $\mathrm{soft}(\mathcal{G}) = \mathcal{G}$ for every DAG, since there are no distinct DAGs with the same transitive closure. For $q = 3$, the only case in which two DAGs have the same transitive closure (up to relabeling the nodes) is the segment DAG $Z_3 \to Z_2 \to Z_1$, denoted by ▬, and the DAG with extra edge $Z_3 \to Z_1$, denoted by ▲. Since ▲ is obtained from ▬ by adding Z_1 to the children of Z_3, we know that ▲ $\in$ soft (▬). On the other hand, since $\mathrm{de}_{\blacktriangle}(j) \setminus \mathrm{ch}_{\blacktriangle}(j) = \emptyset$ for $j = 1, 2$, the only relevant submatrices are obtained for $j = 3$ and they are

$$[(I - \Lambda^{(0)}_{\blacktriangle})^{-1}]_{1,2} \quad \text{and} \quad [(I - \Lambda^{(0)}_{\blacktriangle})^{-1}]_{1,\{2,3\}}$$

for which the rank condition is trivially satisfied, hence ▬ $\in \mathrm{soft}(\blacktriangle)$. Therefore, for DAGs on 3 nodes, the soft-compatible classes are the classes of DAGs with the same transitive closure.

For $q = 4$, if every node has at most 1 descendant that is not a child, the rank condition is always satisfied. The only time this might not hold is for DAGs whose transitive closure is the complete DAG on 4 nodes. There are 8 such DAGs (up to relabeling). We compute their soft-compatible classes.

Example 3.16 Consider DAGs on 4 nodes, with transitive closure the complete DAG



The submatrices of $(I - \Lambda^{(0)}_{\mathcal{G}})^{-1}$ for $j = 1, 2, 3$ are either empty or they have one row, for every $\mathcal{G}'$. Therefore, on these nodes the rank condition is always satisfied. The interesting rank condition comes from the submatrices for $j = 4$. By computing these submatrices for all pairs of DAGs, one obtains

$$\mathrm{soft}(\mathcal{G}_1) = \mathrm{soft}(\mathcal{G}_2) = \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4, \mathcal{G}_5, \mathcal{G}_6, \mathcal{G}_7, \mathcal{G}_8\},$$
$$\mathrm{soft}(\mathcal{G}_3) = \mathrm{soft}(\mathcal{G}_4) = \mathrm{soft}(\mathcal{G}_5) = \mathrm{soft}(\mathcal{G}_6) = \mathrm{soft}(\mathcal{G}_7) = \mathrm{soft}(\mathcal{G}_8) = \{\mathcal{G}_3, \mathcal{G}_4, \mathcal{G}_5, \mathcal{G}_6, \mathcal{G}_7, \mathcal{G}_8\}.$$

For $n = 1, 2$ and $m = 3, \ldots, 8$, we have $\mathcal{G}_n \notin \mathrm{soft}(\mathcal{G}_m)$, since

$$\mathrm{rank}[(I - \Lambda^{(0)}_{\mathcal{G}_m})^{-1}]_{\mathrm{de}_{\mathcal{G}_n}(4) \setminus \mathrm{ch}_{\mathcal{G}_n}(4),\ \mathrm{ch}_{\mathcal{G}_n}(4)} = \mathrm{rank}[(I - \Lambda^{(0)}_{\mathcal{G}_m})^{-1}]_{\{1,2\},3} = 1,$$
$$\mathrm{rank}[(I - \Lambda^{(0)}_{\mathcal{G}_m})^{-1}]_{\mathrm{de}_{\mathcal{G}_n}(4) \setminus \mathrm{ch}_{\mathcal{G}_n}(4),\ \overline{\mathrm{ch}}_{\mathcal{G}_n}(4)} = \mathrm{rank}[(I - \Lambda^{(0)}_{\mathcal{G}_m})^{-1}]_{\{1,2\},\{3,4\}} = 2.$$

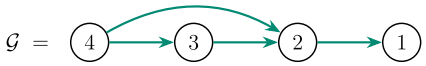
Soft-compatible classes are in general smaller than the DAGs with the same transitive closure. In a soft-compatible class a unique sparsest DAG does not exist, see Example 3.16 where the sparsest DAGs in $\mathrm{soft}(\mathcal{G}_8)$ are $\mathcal{G}_3, \mathcal{G}_4$. Moreover, soft-compatible classes are not equivalence classes, see Example 3.16 where $\mathcal{G}_8 \in \mathrm{soft}(\mathcal{G}_1)$ but $\mathcal{G}_1 \notin \mathrm{soft}(\mathcal{G}_8)$.

3.3.2 More Interventions

So far in Sect. 3 we have mostly assumed one intervention per latent node, and we proved that this is not sufficient to recover the latent DAG or the parameters. It is natural to wonder whether more interventions would allow the recovery. We already discussed after Corollary 3.9 that complete identifiability is impossible regardless of the number of interventions; in other words, Corollary 1.8 holds.

In some cases, increasing the number of interventions allows us to recover more parameters (see Example 3.18). In other cases, more interventions do not improve the parameters that can be recovered (see Example 3.17).

Example 3.17 Consider the DAG



Node $j = 4$ has 2 children and 3 descendants, so $c_4 = 1$. We are interested in the question: if we have more interventions, can we recover more? In particular, can we recover $\lambda_{2,4}^{(0)}, \lambda_{3,4}^{(0)}$?

The only intervention that could give us more information is an extra intervention on node 1, since that is the only descendant of 4 which is not its child. Assume a fifth context also with intervention target 1. We have a linear system with two equations and two unknowns:

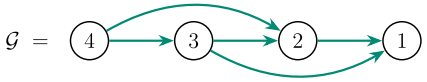
$$\begin{aligned}\Delta_{1,2}^{(1)}\lambda_{2,4}^{(0)} + \Delta_{1,3}^{(1)}\lambda_{3,4}^{(0)} &= \Delta_{1,4}^{(1)},\\ \Delta_{1,2}^{(5)}\lambda_{2,4}^{(0)} + \Delta_{1,3}^{(5)}\lambda_{3,4}^{(0)} &= \Delta_{1,4}^{(5)}.\end{aligned}$$

The matrix of this linear system, however, has rank 1 because

$$\begin{pmatrix}\Delta_{1,3}^{(1)}\\ \Delta_{1,3}^{(5)}\end{pmatrix} = \lambda_{2,3}^{(0)}\begin{pmatrix}\Delta_{1,2}^{(1)}\\ \Delta_{1,2}^{(5)}\end{pmatrix} \qquad \text{and} \qquad \begin{pmatrix}\Delta_{1,4}^{(1)}\\ \Delta_{1,4}^{(5)}\end{pmatrix} = (\lambda_{2,4}^{(0)} + \lambda_{2,3}^{(0)}\lambda_{3,4}^{(0)})\begin{pmatrix}\Delta_{1,2}^{(1)}\\ \Delta_{1,2}^{(5)}\end{pmatrix}.$$

All paths from 4 to 1 must go through 2 and this means no additional parameters can be recovered from the extra intervention.

Example 3.18 Consider the DAG



and focus on the node $j = 4$. The linear system is as above, but now

$$\begin{pmatrix} \Delta^{(1)}_{1,3} \\ \Delta^{(5)}_{1,3} \end{pmatrix} \nparallel \begin{pmatrix} \Delta^{(1)}_{1,2} \\ \Delta^{(5)}_{1,2} \end{pmatrix},$$

so the matrix is full rank and we can recover the parameters $\lambda^{(0)}_{2,4}, \lambda^{(0)}_{3,4}$.

To conclude, more interventions do not necessarily allow us to recover the DAG or the parameters. Indeterminates $\lambda^{(k)}_{i,j}$ with $\mathrm{ch}(i) = \emptyset$ are not the only ones that cannot be recovered regardless of how many soft interventions we have - this is also the case for $\lambda^{(k)}_{2,4}$ and $\lambda^{(k)}_{3,4}$ in Example 3.17. But there are examples where more interventions reduce the dimension of the solution space - we can recover $\lambda^{(k)}_{2,4}$ and $\lambda^{(k)}_{3,4}$ in Example 3.18.

4 Algorithm

Coupled tensor decomposition of higher-order cumulants enables us to recover parameters in an LCD model. In this section, we explain how to turn our results into a numerical algorithm for LCD. The algorithm has two main steps. The first (see Sect. 4.1) is to identify intervention targets, permutation, and scaling. This turns the results of Sect. 2 into an algorithm, and works for both perfect and soft interventions. The second step (see Sect. 4.2) is to recover the parameters of the model. We do this step for perfect interventions (following Theorem 1.6) and not for soft interventions, in light of Theorem 1.7. Both steps simplify when $q \leq p$; that is, when the number of latent variables is at most the number of observed variables, see Sect. 4.3. We test our algorithms on synthetic data in Sect. 4.4.

4.1 Recovery of Intervention Targets, Permutation, and Scaling

The algorithm input is the d-th order cumulants $\kappa_d(X^{(k)})$ as in (10), for $k \in K \cup \{0\}$ ranging over contexts and a fixed $d \geq 3$. The input tensors are either exact (population cumulants) or approximate (sample cumulants).

For our fixed d, we assume that the decomposition in (10) is unique, that $\kappa_d(\varepsilon^{(k)}_i) \neq 0$ for all k and all $i \in [q]$, and that $\kappa_d(\varepsilon^{(k)}_{i_k}) \neq \pm 1$ for all k. This is the same assumption as appears temporarily in the proof of Proposition 2.3. The assumption is not necessary, since one can combine information from multiple higher-order cumulants, but it helps our exposition, as in the proof of Proposition 2.3. In our experiments, we consider $d = 3$ and $d = 4$.

Tensor decomposition recovers the matrices $A^{(k)}$ up to permutation and scaling, by Proposition 2.1. Tensor decomposition thus recovers a set of matrices

$\{A^{(k)}D^{(k)}P^{(k)} : k \in K \cup \{0\}\}$ for unknown scaling matrices $D^{(k)}$ and unknown permutations $P^{(k)}$. In practice, any numerical tensor decomposition algorithm can be used for this step. We use the subspace power method [26] or simultaneous diagonalization [19] when $q \leq p$. We can assume without loss of generality that $D^{(0)} = P^{(0)} = I$, see Proposition 2.5. Then the other scaling matrices $D^{(k)}$ have all entries ± 1 except one, by (9) of Proposition 2.3. There is one entry that is not ± 1, which corresponds to the intervention target of context k.

To find the permutation, we consider the difference $A^{(0)} - A^{(k)}D^{(k)}P^{(k)}$, as $D^{(k)}$ varies over diagonal matrices with diagonal ± 1 and $P^{(k)}$ varies over permutation matrices. That is, the product $D^{(k)}P^{(k)}$ is a signed permutation matrix. The rank of the difference equals one if and only if we have the correct sign and order, see Corollary 2.10. This suggests an algorithm: for all $q \times q$ signed permutation matrices Q (of which there are $2^q \times q!$) compute $A^{(k)}D^{(k)}P^{(k)}Q$ and compute the second largest eigenvalue of the matrix $A^{(0)} - A^{(k)}D^{(k)}P^{(k)}Q$. Choose Q for which this eigenvalue is smallest. This Q is $(P^{(k)})^\top$, up to sign, where the sign gives all but entry $D^{(k)}_{i_k,i_k}$ of $D^{(k)}$. See Algorithm 1.

To find the remaining entry of $D^{(k)}$, we compare the columns of $A^{(0)}$ and $A^{(k)}D^{(k)}$. The only column that differs between the two matrices is the i_k-th column. The i_k-th columns of the two matrices are collinear, with scaling $D^{(k)}_{i_k,i_k}$. This recovers the intervention target i_k and the scaling matrix $D^{(k)}$. See Algorithm 2.

A faster way approach to find the intervention targets, permutation, and scaling could be to implement Proposition 2.7. One can compare each column of $A^{(0)}$ to each column of $A^{(k)}D^{(k)}P^{(k)}$, e.g. by projecting a column v_1 of one matrix onto another column v_2 of the other. Each time, the residue of the projection $\|\mathrm{v}_1 - \pi_{\langle \mathrm{v}_2 \rangle}(\mathrm{v}_1)\|$ and the scaling $\frac{1}{\|\mathrm{v}_2\|}\|\pi_{\langle \mathrm{v}_2 \rangle}(\mathrm{v}_1)\|$ can be stored and thresholds can be used to decide which numerical values are zero or ± 1. Such a procedure is numerically sensitive and influenced by the threshold. The threshold determines the assignment of intervention target, and we want the $|K|$ intervention targets to cover all latent nodes. One must choose a threshold that gives such an assignment. We leave this for future work.

4.2 Recovery of Parameters

Once the intervention targets, permutation, and scaling are recovered, using Sect. 4.1, we have matrices $A^{(k)} = F(I - \Lambda^{(k)})^{-1}$ for $k = 0, \ldots, q$. We can relabel contexts so that the intervention target of the k-th context is k. We construct $(I - \Lambda^{(0)})^{-1}$ as

$$(I - \Lambda^{(0)})^{-1}_{i,j} = \begin{cases} 1 & i = j, \\ \frac{A^{(0)}_{1,j} - A^{(i)}_{1,j}}{A^{(0)}_{1,i}} & i \neq j, \end{cases}$$

following the proof of Theorem 1.6. We invert this matrix to find $\Lambda^{(0)}$. This is Algorithm 3. Finally, we recover F using $F = A^{(0)}(I - \Lambda^{(0)})$. One can compare the products $A^{(k)}(I - \Lambda^{(k)})$ for different contexts k to test the goodness of fit of the LCD model. In theory, these should all return the same mixing matrix F.

4.3 The Injective Case

Restricting to the case $q \leq p$ allows for simplifications, cf. Remarks 2.11, 3.5, and 3.12. First, the tensor decomposition step can achieved using simultaneous diagonalization [19]. We explain how to recover the intervention targets, permutation, and scaling in this setting. When $q \leq p$ the Moore-Penrose pseudo-inverse satisfies

$$C^{(k)} := \left(F(I - \Lambda^{(k)})^{-1} D^{(k)} P^{(k)}\right)^{+} = (P^{(k)})^{\top} (D^{(k)})^{-1} (I - \Lambda^{(k)}) H, \quad (24)$$

where $H = F^{+}$. In particular, $C^{(0)} = (I - \Lambda^{(0)})H$. Let $(\mathrm{c}^{(k)})^{\ell}$ denote the ℓ-th row of $C^{(k)}$. Then we have the following result, in the same spirit as Proposition 2.7.

Proposition 4.1 *Consider LCD under Assumption 1.2 where $q \leq p$. Fix $k \in K$ and let σ be the permutation associated to the permutation matrix $P^{(k)}$. Then*

$$(\mathrm{c}^{(0)})^{\ell} = (\mathrm{c}^{(k)})^{\sigma(\ell)}$$

if and only if $\ell \neq i_k$, where i_k is the target of the k-th intervention.

Proof We have formulae

$$(\mathrm{c}^{(0)})^{\ell} = \mathrm{h}^{\ell} - \sum_{j \in \mathrm{pa}(\ell)} \lambda^{(0)}_{\ell,j} \mathrm{h}^{j}, \qquad (\mathrm{c}^{(k)})^{\sigma(\ell)} = \frac{1}{D^{(k)}_{\ell,\ell}} \left(\mathrm{h}^{\ell} - \sum_{j \in \mathrm{pa}(\ell)} \lambda^{(k)}_{\ell,j} \mathrm{h}^{j} \right),$$

by (24). If $\ell \neq i_k$, then $D^{(k)}_{\ell,\ell} = 1$ and $\lambda^{(k)}_{\ell,j} = \lambda^{(0)}_{\ell,j}$ for all $j \in [q]$. Hence these two expressions coincide. If $\ell = i_k$ then under the genericity assumption we have $(\mathrm{c}^{(0)})^{\ell} \neq (\mathrm{c}^{(k)})^{\sigma(\ell)}$. □

Proposition 4.1 enables us to find the intervention target i_k and the permutation matrix $P^{(k)}$, as follows. For sufficiently general parameters $\Lambda^{(0)}$ and $\Lambda^{(k)}$ and error distribution $\varepsilon^{(k)}$, the rows $(\mathrm{c}^{(0)})^{i_k}$ and $(\mathrm{c}^{(k)})^{\sigma(i_k)}$ differ. Hence, i_k is the index of the row of $C^{(0)}$ without a match in $C^{(k)}$. We recover $P^{(k)}$ by matching the remaining rows: if $\ell \neq i_k$, then there exists j such that $(\mathrm{c}^{(0)})^{\ell} = (\mathrm{c}^{(k)})^{j}$. This finds all but one row of $P^{(k)}$. It has a unique completion to a permutation matrix, which matches $(\mathrm{c}^{(0)})^{i_k}$ to the row of $C^{(k)}$ without a match in $C^{(0)}$. See Algorithms 4 and 5. Algorithm 6 recovers the scalings $D^{(k)}$.

We now explain how to recover the parameters. In light of the above, we have matrices $(A^{(k)})^{+} = (I - \Lambda^{(k)})H$. We find $H = F^{+}$ as follows. The i_k-th row of $(A^{(k)})^{+}$ has entries

$$(A^{(k)})^{+}_{i_k,j} = \sum_{\ell \in [q]} (I - \Lambda^{(k)})_{i_k,\ell} H_{\ell,j} = H_{i_k,j}.$$

This is Algorithm 7. To find $\Lambda^{(0)}$, we use

$$(A^{(0)})^{+}H^{+} = (I - \Lambda^{(0)})HH^{+} = (I - \Lambda^{(0)}).$$

Following the initial tensor decomposition, the time complexity of this algorithm is determined by the time required for the alignment and to calculate the pseudo-inverses of the products. The former takes time $O(q^3p)$ and the latter $O(p^2q^2)$, so the overall runtime is $O(q^2p\max(p,q))$. This improves on the algorithm in [50] provided we ignore the time taken to construct and decompose the higher-order cumulants.

4.4 Numerical Experiments

We test our algorithms on synthetic data. The general procedure for any (p, q) is implemented in Algorithms 1-3 in Appendix A and the injective case ($q \leq p$) is implemented in Algorithms 4-7. For the general setting, we use $d = 4$, since we use the subspace power method for tensor decomposition [26] and it requires input of even order. We use the subspace power method because it can be used to decompose any even-order symmetric real tensor with low rank, it outperforms other state-of-the-art methods, and is robust to noisy input (see [26] for details). For the injective case, we study $d = 3$. These choices of d satisfy Assumption 1.2(c), since we assume error distributions that are sufficiently general to have non-vanishing third or fourth order cumulants, for instance they are not Gaussian or symmetrically distributed. A larger d is needed in Assumption 1.2(c) only in the special case that some higher-order cumulants of the error distributions vanish.

We sample graphs using the Python package |causaldag| [49]. It extends the Erdős–Rényi model [43] to DAGs: given an edge density ρ, the edge $i \to j$ is added to the graph with probability ρ, and if and only if $i > j$. We fix $\rho = 0.75$, sample the entries of F^{+} independently from Unif($[-2, 2]$), and the non-zero entries of $\Lambda^{(0)}$ independently from Unif($\pm[0.25, 1]$), as in [50]. We fix $p = 5$ and vary q from 2 to 7. For each value of q, we generate 500 models and calculate the relative Frobenius error for the recovery of F and $\Lambda^{(0)}$, which is $\frac{1}{\|M\|}\|\widetilde{M} - M\|$, where $\|\cdot\|$ denotes the Frobenius norm, M is the true matrix, and $\widetilde{M}$ is the recovered matrix. We also calculate DAG recovery error, as follows. A penalty of 1 is incurred if the algorithm recovers a non-existent edge or misses an existing one, while recovering an edge in the wrong direction incurs a penalty of 2. Then we sum the penalty over all edges.

We plot the median error in recovering F, $\Lambda^{(0)}$ in Fig. 2. A significant portion of the error is due to the tensor decomposition step, especially in the case $q > p$, but improved algorithms for tensor decomposition are beyond the scope of this paper. Therefore, together with the error of the recovery starting from the population cumulants $\kappa_d(X^{(k)})$, we display also the error of the recovery obtained directly from the factor matrices $A^{(k)}D^{(k)}P^{(k)}$, as if these had been recovered perfectly from tensor decomposition.

For the injective case, we fix $p = 10$ and let q vary from 2 to 10. We consider 500 models generated as above, and compute the relative Frobenius error for both recovered matrices F and $\Lambda^{(0)}$. The median errors are plotted in Fig. 3, starting from the cumulants (blue graph) as well as from the factor matrices (orange graph).

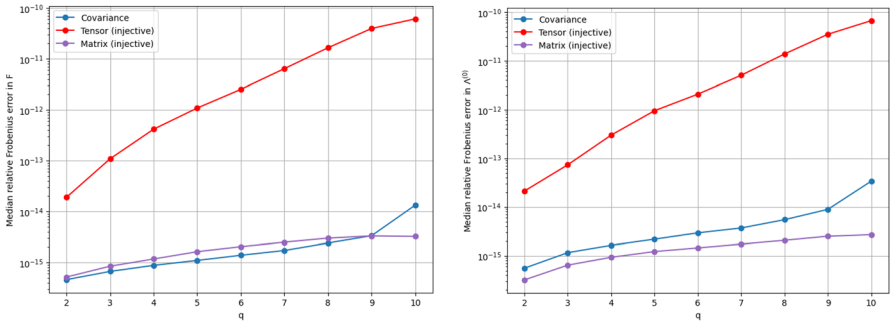


Fig. 3 Median relative Frobenius error in the recovery of F (left) and $\Lambda^{(0)}$ (right) when $p = 10$, using the injective algorithm. Note the logarithmic scale on the y-axis

We explain how our algorithm for LCD can be used in practice, and what it finds. Assume we have $K + 1$ contexts: an observational context and K other contexts. While the K contexts may result in complicated changes to the distribution over the observed variables, the idea of LCD is to build a latent representation such that they can be viewed as a single-node intervention on the latent variables, and such that each context is an intervention on a different latent variable. Each latent variable is a linear combination of observed variables. For example, if the observed variables correspond to genes and the contexts to mutations, then the latent variables are weighted combinations of genes, which can be thought of as weighting the effect of a mutation on each gene. See [50] for a workflow of LCD on a single-cell RNA sequencing dataset. For LCD, the input consists of $n_0 + \cdots + n_K$ data points, where n_k data points are observed in context k. We build the d-th order sample cumulant for each context k. Any $d \geq 3$ can be used, since for sufficiently general data the higher-order cumulants of the error distributions will not vanish. We run our LCD algorithm on the tuple of cumulant tensors. We use the general formulation in Algorithms 1-3. If $K \leq p$ we can also use the injective approach in Algorithms 4-7. The output is a tuple of K latent variables, together with estimates for the latent graph $\mathcal{G}$, weights Λ, and the linear mixing map F.

5 Outlook

We have studied the identifiability of linear causal disentanglement using tensor decomposition of higher-order cumulants. We view the parameters compatible with a given model as the solution space to a system of equations. Identifiability holds when the space has dimension zero, and can be achieved using perfect interventions. Here, we give an algorithm to recover the parameters. For soft interventions, we recover a compatibility class of graphs and parameters. We conclude with some open problems for future investigation.

On the theoretical side, the first question is of combinatorial nature. The definition of $\mathrm{soft}(\mathcal{G})$ in Definition 3.14 involves ranks of matrices. These rank conditions encode information about the paths in $\mathcal{G}$. This suggests the following problem.

Problem 5.1 Find a combinatorial description of $\mathrm{soft}(\mathcal{G})$, based on the structure of $\mathcal{G}$.

Higher-order cumulants can reduce the degree of the solution space of parameters, as compared to covariance matrices, see Proposition 3.3. An open problem is whether they restricts the set of compatible DAGs.

Problem 5.2 Let $\mathrm{soft}_2(\mathcal{G})$ denote the graphs $\mathcal{G}'$ for which there exist parameters $F, \Lambda^{(k)}$ defined according to $\mathcal{G}'$ such that the covariance matrix coincides with the covariance matrix of a model with latent DAG $\mathcal{G}$. Are the following containments strict in general

$$\mathrm{soft}(\mathcal{G}) \subset \mathrm{soft}_2(\mathcal{G}) \subset \{\mathcal{G}' \mid \overline{\mathcal{G}'} = \overline{\mathcal{G}}\}?$$

Under soft interventions, the space of parameters compatible with a given model is linear and positive dimensional, by Theorem 1.7. It is then natural to ask for the best solution in this space, for an appropriate notion of best. This would give a choice of unique parameters under soft interventions.

Finally, our assumptions require that *all* the latent error distributions are non-Gaussian. The results may extend to the case where some are Gaussian, cf. [60].

On the algorithmic side, there are multiple possible improvements. The tensor decomposition contributes significantly to the error in the recovered parameters, see Fig. 2. Other tensor decomposition algorithms might give more accurate output. One could test our algorithm starting from the factor matrices plus random noise, to study the extent to which our algorithm would work with a sufficiently accurate tensor decomposition. Next, one could implement a greedy search over permutations and signs to speed up the recovery of $P^{(k)}$ and $D^{(k)}$. Finally, it would be interesting to study the robustness to non-linearity in the latent space (e.g., $Z = (I - \Lambda)^{-1}\varepsilon + \alpha\varepsilon^2$ for small $\alpha \in \mathbb{R}$) or in the mixing map (e.g., $X = FZ + \alpha Z^2$, where Z^2 is a vector with entries $Z_i Z_j$ for all $i, j \in [q]$ and $\alpha \in \mathbb{R}$ small).

A Pseudocode

We provide pseudocode for the algorithms in Sect. 4. Their implementations are available at https://github.com/paulaleyes14/linear-causal-disentanglement-via-cumulants. Below, the i-th row of a matrix M is denoted by m^i and the i-th column by m_i.

Algorithms 3 and 7 below work if the set of intervention targets coincides with the set of latent variables. In theory, this is true by our assumptions. In practice, the algorithm could assign the wrong target to an intervention due to numerical errors. When implementing the algorithm, we force the interventions to be on distinct nodes.

A.1 General Case

Input: $M = A^{(0)}$ and $M^{(k)} = A^{(k)} D^{(k)} P^{(k)}$.
Output: $P^{(k)}$, the permutation matrix encoding the relabeling of the latent nodes in the context corresponding to an intervention at node i_k.

$q \leftarrow$number of columns of M
perm $\leftarrow$ set of all $q \times q$ permutation matrices with entries ± 1
$\sigma \leftarrow$ maximum float
$P \leftarrow$ None
for mat in perm **do**
    newmat $\leftarrow M - M^{(k)} \cdot$ mat
    $ss \leftarrow$ second largest abs(singular value) of newmat
    if $ss < \sigma$ **then**
        $\sigma \leftarrow ss$
        $P \leftarrow \text{mat}^\top$
    else
        continue
    end if
end for
return P

Algorithm 1 Recovery of the permutation matrix (recover_perm)

Input: $M = A^{(0)}$, $M^{(k)} = A^{(k)} D^{(k)} P^{(k)}$ and a threshold `thr`.
Output: the target of the k-th intervention i_k and the diagonal matrix $D^{(k)}$.

$q \leftarrow$ number of columns of M
$D \leftarrow I_{q \times q}$
$P \leftarrow \text{recover\_perm}(M, M^{(k)})$
$N \leftarrow M^{(k)} P^\top$
scalings $\leftarrow$ list()
indices $\leftarrow$ list()
for $i = 1$ to q **do**
    $v \leftarrow$ project $\mathbf{n}^i$ onto $\mathbf{m}^i$
    if $|v - \mathbf{n}^i| <$ `thr` **then**
        Add $v[1]/\mathbf{m}^i[1]$ to scalings
        Add i to indices
    end if
end for
$d \leftarrow$ largest entry of scalings
$i_k \leftarrow$ indices(index of d)
$D[i_k, i_k] \leftarrow d$
return i_k, D

Algorithm 2 Recovery of the intervention target and scaling (recover_target_scaling)

Input: $M = A^{(0)}$, $M^{(k)} = A^{(k)} D^{(k)} P^{(k)}$ for all $k \in [q]$, and a threshold `thr`.
Output: $\Lambda^{(0)}$.

$q \leftarrow$ number of columns of M
$L \leftarrow I_{q \times q}$
for $k = 1$ to q **do**
    $P \leftarrow \text{recover\_perm}(M, M^{(k)})$
    $(i_k, D) \leftarrow \text{recover\_target\_scaling}(M, M^{(k)}, \texttt{thr})$
    $A = M^{(k)}\ P^{\top}\ \text{inverse}(D)$
    for $j = 1$ to q **do**
        if $j \neq i_k$ **then**
            $L[i_k, j] \leftarrow \frac{M[1,j] - A[1,j]}{M[1,i_k]}$
        end if
    end for
end for
return $I_{q \times q} - \text{inverse}(L)$

Algorithm 3 Recovery of $\Lambda^{(0)}$ (recover_Lambda)

A.2 Injective Case

Input: $C = C^{(0)}$ and $C^{(k)} = (P^{(k)})^{\top} (D^{(k)})^{-1} (I - \Lambda^{(k)}) H$.
Output: (i_k, j_k) such that i_k is the intervention target of the k-th context, and $P^{(k)}_{i_k, j_k} = 1$.

$q \leftarrow$number of rows of C
$\text{matched}_{\text{obs}} \leftarrow \text{set}()$
$\text{matched}_{\text{int}} \leftarrow \text{set}()$
for $i = 1$ to q **do**
    if $\mathbf{c}^i$ has matching row in $C^{(k)}$ **then**
        $j \leftarrow$ index of matching row
        Add i to $\text{matched}_{\text{obs}}$
        Add j to $\text{matched}_{\text{int}}$
    end if
end for
$i_k \leftarrow [q] \backslash \text{matched}_{\text{obs}}$
$j_k \leftarrow [q] \backslash \text{matched}_{\text{int}}$
return (i_k, j_k)

Algorithm 4 Recovery of the intervention target (recover_target)

Input: $C = C^{(0)}$ and $C^{(k)}$, recovered from the context with intervention target i_k.
Output: the permutation matrix $P^{(k)}$.

$q \leftarrow$number of rows of C
$P \leftarrow \mathbf{0}_{q \times q}$
$(i_k, j_k) \leftarrow \text{recover\_target}(C, C^{(k)})$
$P[i_k, j_k] \leftarrow 1$
for $i = 1$ to q **do**
  if $i = i_k$ **then**
    continue
  else
    $j \leftarrow$ index of matching row in $C^{(k)}$ to $\mathbf{c}^i$
    $P[i, j] \leftarrow 1$
  end if
end for
return P

Algorithm 5 Recovery of the permutation matrix (recover_perm)

Input: $C = C^{(0)}$ and $C^{(k)}$.
Output: the diagonal matrix $D^{(k)}$.

$q \leftarrow$number of rows of C
$D \leftarrow I_{q \times q}$
$(i_k, j_k) \leftarrow \text{recover\_target}(C, C^{(k)})$
$P \leftarrow \text{recover\_perm}(C, C^{(k)})$
$B^{(0)} \leftarrow \text{pseudoinverse}(PC^{(0)})$
$B^{(k)} \leftarrow \text{pseudoinverse}(PC^{(k)})$
$v \leftarrow$ project $\mathbf{b^{(k)}}_{i_k}$ onto $\mathbf{b^{(0)}}_{i_k}$
$D[i_k, i_k] \leftarrow v[1]/\mathbf{b^{(0)}}_{i_k}[1]$

return D

Algorithm 6 Recovery of the scaling (recover_scaling)

Input: $C = C^{(0)}$ and $C^{(k)}$ for all $k \in [q]$.
Output: F such that $C^{(0)} = (I - \Lambda^{(0)})H$ and $H = F^{+}$.

$q \leftarrow$ number of rows in C
for $k = 1$ to q **do**
    $(i_k, j_k) \leftarrow \text{recover\_target}(C, C^{(k)})$
    $P \leftarrow \text{recover\_perm}(C, C^{(k)})$
    $D \leftarrow \text{recover\_scaling}(C, C^{(k)})$
    $A = PDC^{(k)}$
    $\mathbf{h}^{i_k} = \mathbf{a}^{i_k}$
end for
return pseudoinverse(H)

Algorithm 7 Recovery of $F = H^{+}$ (recover_F)

Funding Open access funding provided by Swiss Federal Institute of Technology Zurich

Declarations

Competing interests CM was supported by Dr. Max Rössler, the Walter Haefner Foundation and the ETH Zürich Foundation. AS was partially supported by the NSF (DMR- 2011754).

References

1. Anandkumar, A., Ge, R., Hsu, D.J., Kakade, S.M., Telgarsky, M., et al.: Tensor decompositions for learning latent variable models. J. Mach. Learn. Res. **15**(1), 2773–2832 (2014)
2. Ahuja, K., Hartford, J., Bengio, Y.: An equivariance perspective on identifiable representation learning. Properties from mechanisms (2021)
3. Adams, J., Hansen, N., Zhang, K.: Identification of partially observed linear causal models: Graphical conditions for the non-Gaussian and heterogeneous cases. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, Advances in Neural Information Processing Systems, volume 34, pages 22822–22833. Curran Associates, Inc., (2021)
4. Ahuja, K., Mahajan, D., Wang, Y., Bengio, Y.: Interventional causal representation learning. In: Proceedings of the 40th International Conference on Machine Learning, ICML'23. JMLR.org, (pp. 372–407) (2023)
5. Bengio, Y., Courville, A., Vincent, P.: A review and new perspectives. Represent. Learn. (2014)
6. Olivier Berne, C., Joblin, Y.D., Smith, J.D., Rapacioli, M., Bernard, J.P., Thomas, J., Reach, W., Abergel, A.: Analysis of the emission of very small dust particles from spitzer spectro-imagery data using blind signal separation methods. Astron. Astrophys. **469**(2), 575–586 (2007)

7. Bollen, K.A.: Structural Equations with Latent Variables, volume 210. John Wiley & Sons (1989)
8. Buchholz, S., Rajendran, G., Rosenfeld, E., Aragam, B., Schölkopf, B., Ravikumar, P.: Learning linear causal representations from interventions under general nonlinear mixing. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems, vol. 36, pp. 45419–45462. Curran Associates Inc (2023)
9. Bing, S., Wahl, J., Ninad, U., Runge, J.: Invariance & causal representation learning: Prospects and limitations. In: Causal Representation Learning Workshop at NeurIPS 2023 (2023)
10. Comon, P., Jutten, C.: Handbook of Blind Source Separation: Independent component analysis and applications. Academic Press (2010)
11. Comon, P.: Independent component analysis, a new concept? Signal Process. **36**, 287–314 (1994)
12. Chiantini, L., Ottaviani, G., Vannieuwenhoven, N.: On generic identifiability of symmetric tensors of subgeneric rank. Trans. Am. Math. Soc. **369**(6), 4021–4042 (2017)
13. Cai, R., Xie, F., Glymour, C., Hao, Z., Zhang, K.: Triad constraints for learning causal structure of latent variables. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc. (2019)
14. Decker, W., Eder, C., Fieker, C., Horn, M., Joswig, M. (eds): The Computer Algebra System OSCAR: Algorithms and Examples. Algorithms and Computation in Mathematics. Springer (Vol. 32) (2024)
15. De Lathauwer, L., Castaing, J., Cardoso, J.-F.: Fourth-order cumulant-based blind identification of underdetermined mixtures. IEEE Trans. Signal Process. **55**(6), 2965–2973 (2007)
16. Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C.P., Jerby-Arnon, L., Marjanovic, N.D., Dionne, D., Burks, T.: Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. Cell **167**(7), 1853–1866 (2016)
17. Eberhardt, F., Glymour, C., Scheines, R.: On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables. In: Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence, pp. 178–184 (2005)
18. Eriksson, J., Koivunen, V.: Identifiability, separability, and uniqueness of linear ICA models. IEEE Signal Process. Lett. **11**(7), 601–604 (2004)
19. Harshman, R.A.: Foundations of the parafac procedure: models and conditions for an "explanatory" multimodal factor analysis. UCLA Work. Pap. Phonet. **16**, 1–84 (1970)
20. Halpern, Y., Horng, S., Sontag, D.: Anchored discrete factor analysis. arXiv:1511.03299 (2015)
21. Hyvärinen, A., Pajunen, P.: Nonlinear independent component analysis: existence and uniqueness results. Neural Netw. **12**, 429–439 (1999)
22. Jiang, Y., Aragam, B.: Learning nonparametric latent causal graphs with unknown interventions. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems, vol. 36, pp. 60468–60513. Curran Associates Inc (2023)
23. Jung, T.-P., Makeig, S., McKeown, M.J., Bell, A.J., Lee, T.-W., Sejnowski, T.J.: Imaging brain dynamics using independent component analysis. Proc. IEEE **89**(7), 1107–1122 (2001)
24. Jutten, C.: Calcul neuromimétique et traitement du signal: analyse en composantes indépendantes. PhD thesis, Grenoble INPG (1987)
25. Khemakhem, I., Kingma, D., Monti, R., Hyvarinen, A.: Variational autoencoders and nonlinear ICA: a unifying framework. In: International Conference on Artificial Intelligence and Statistics, pp. 2207–2217. PMLR (2020)
26. Kileel, J., Pereira, J.M.: Subspace power method for symmetric tensor decomposition and generalized PCA. arXiv preprint arXiv:1912.04007v4 (2024)
27. Kivva, B., Rajendran, G., Ravikumar, P., Aragam, B.: Learning latent causal graphs via mixture oracles (2021)
28. Kekić, A., Schölkopf, B., Besserve, M.: Targeted reduction of causal models. In: ICLR 2024 Workshop on AI4DifferentialEquations in Science (2024)
29. Landsberg, J.M.: Tensors: Geometry and Applications, vol. 128. American Mathematical Soc, Providence (2011)
30. Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schökopf, B., Bachem, O.: Challenging common assumptions in the unsupervised learning of disentangled representations (2019)
31. Lachapelle, S., Rodríguez López, P., Sharma, Y., Everett, K., Priol, R.L., Lacoste, A., Lacoste-Julien, S.: Nonparametric partial disentanglement via mechanism sparsity: sparse actions, interventions and sparse temporal dependencies (2024)

32. Lachapelle, S., Mahajan, D., Mitliagkas, I., Lacoste-Julien, S.: Additive decoders for latent variables identification and cartesian-product extrapolation. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems, vol. 36, pp. 25112–25150. Curran Associates Inc (2023)
33. Liu, Y., Zhang, Z., Gong, D., Gong, M., Huang, B.,Hengel, A.V.D., Zhang, K., Shi, J.Q.: Identifying weight-variant latent causal models. arXiv:2208.14153 (2023)
34. Liu, Y., Zhang, Z., Gong, D., Gong, M.: Identifiable latent neural causal models. Anton van den Hengel (2024)
35. Liu, Y., Zhang, Z., Gong, D., Gong, M., Huang, B., Hengel, A.V.D., Zhang, K., Qinfeng Shi, J.: Identifiable latent polynomial causal models through the lens of change. In: The Twelfth International Conference on Learning Representations (2024)
36. Grayson D.R., Stillman, M.E.: Macaulay2, a software system for research in algebraic geometry. Available at http://www2.macaulay2.com (2002)
37. Marcinkiewicz, J.: Sur une propriété de la loi de Gauss. Math. Z. **44**(1), 612–618 (1939)
38. McCullagh, P.: Tensor methods in statistics: Monographs on statistics and applied probability. Chapman and Hall/CRC (2018)
39. Meinshausen, N., Hauser, A., Mooij, J.M., Peters, J., Versteeg, P., Bühlmann, P.: Methods for causal inference from gene perturbation experiments and validation. Proc. Natl. Acad. Sci. **113**(27), 7361–7368 (2016)
40. Moran, G.E., Sridhar, D., Wang, Y., Blei, D.: Identifiable deep generative models via sparse decoding, Trans. Mach. Learn. Res. (2022)
41. OSCAR – Open Source Computer Algebra Research system, Version 1.0.0. Available at https://www.oscar-system.org (2024)
42. Pearl, J.: Models, Reasoning and Inference. Cambridge, UK: Cambridge University Press **19**(2), 3 (2000)
43. Renyi, E.: On random graph. Publicationes Mathematicate **6**, 290–297 (1959)
44. Sánchez, B.N., Budtz-Jørgensen, E., Ryan, L.M., Howard, H.: Structural equation models: a review with applications to environmental epidemiology. J. Am. Stat. Assoc. **100**(472), 1443–1455 (2005)
45. Shimizu, S.: Lingam: Non-Gaussian methods for estimating causal structures. Behaviormetrika **41**, 65–98 (2014)
46. Schölkopf, B., Locatello, F., Bauer, S., Ke, N.R., Kalchbrenner, N., Goyal, A., Bengio, Y.: Toward causal representation learning. Proc. IEEE **109**(5), 612–634 (2021)
47. Shamsaie, K., Megas, S., Asadollahzadeh, H., Teichmann, S.A., Lotfollahi, M.: Disentangling covariates to predict counterfactuals for single-cell data (2024)
48. Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D.A., Nolan, G.P.: Causal protein-signaling networks derived from multiparameter single-cell data. Science **308**(5721), 523–529 (2005)
49. Squires, C.: causaldag: creation, manipulation, and learning of causal models. https://github.com/uhlerlab/causaldag (2018)
50. Squires, C., Seigal, A., Bhate, S., Uhler, C.: Linear causal disentanglement via interventions. In: Proceedings of the 40th International Conference on Machine Learning, ICML'23. JMLR.org (2023)
51. Silva, R., Scheine, R., Glymour, C., Spirtes, P.: Learning the structure of linear latent variable models. J. Mach. Learn. Res. **7**(8), 191–246 (2006)
52. Sullivant, S.: Algebraic Statistics, volume 194. American Mathematical Society (2018)
53. Triantafillou, S., Lagani, V., Heinze-Deml, C., Schmidt, A., Tegner, J., Tsamardinos, I.: Predicting causal relationships from biological data: applying automated causal discovery on mass cytometry data of human immune cells. Sci. Rep. **7**(1), 12724 (2017)
54. Varici, B., Acarturk, E., Shanmugam, K., Abhishek, K., Tajer, A.: Score-based causal representation learning with interventions (2023)
55. Varici, B., Acartürk, E., Shanmugam, K., Tajer, A.: Score-based causal representation learning from interventions: nonparametric identifiability. In Causal Representation Learning Workshop at NeurIPS 2023 (2023)
56. Varici, B., Acartürk, E., Shanmugam, K., Tajer, A.: General identifiability and achievability for causal representation learning. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, Proceedings of The 27th International Conference on Artificial Intelligence and Statistics, volume 238 of Proceedings of Machine Learning Research, pp. 2314–2322. PMLR, 02–04 May (2024)

57. von Kügelgen, J., Besserve, M., Wendong, L., Gresele, L., Kekić, A.,Bareinboim, E., Blei, D., Schölkopf B.: Nonparametric identifiability of causal representations from unknown interventions. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, Advances in Neural Information Processing Systems, volume 36, pages 48603–48638. Curran Associates, Inc., (2023)
58. Verma, T.S., Pearl J.: Equivalence and synthesis of causal models. In: Probabilistic and Causal Inference: The Works of Judea Pearl, pp. 221–236. Association for Computing Machinery, New York, NY, USA, 1 edition (2022)
59. Wang, Y.S., Drton, M.: High-dimensional causal discovery under non-Gaussianity. Biometrika **107**(1), 41–59 (2020)
60. Wang, K., Seigal, A.: Identifiability of overcomplete independent component analysis. arXiv:2401.14709 (2024)
61. Xie, F., Cai, R., Huang, B., Glymour, C., Hao, Z., Zhang, K.: Generalized independent noise condition for estimating latent variable causal graphs. Adv. Neural. Inf. Process. Syst. **33**, 14891–14902 (2020)
62. Xi, J., Hartford, J.: Propensity score alignment of unpaired multimodal data (2024)
63. Xie, F., Huang, B., Chen, Z., He, Y., Geng, Z., Zhang, K.: Identification of linear non-Gaussian latent hierarchical structure. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pp. 24370–24387. PMLR, 17–23 July (2022)
64. Yang, M., Liu, F., Chen, Z., Shen, X., Hao, J., Wang, J.: CausalVAE: disentangled representation learning via neural structural causal models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9593–9602 (2021)
65. Zhang, J., Squires, C., Greenewald, K., Srivastava, A., Shanmugam, K., Uhler, C.: Identifiability guarantees for causal disentanglement from soft interventions. arXiv:2307.06250 (2023)
66. Zimmermann, R.S., Sharma, Y., Schneider, S., Bethge, M., Brendel, W.: Contrastive learning inverts the data generating process (2022)
67. Zhang, K., Xie, S., Ng, I., Zheng, Y.: Causal representation learning from multiple distributions: a general setting (2024)