# Joint Causal Structure and Cluster Discovery Using Variational Inference

**Avni Rajpal** *ai21resch04001@iith.ac.in*
*Department of Artificial Intelligence, Indian Institute of Technology Hyderabad, India*

**Anubhav Kumar** *ai24mtech12004@iith.ac.in*
*Department of Artificial Intelligence, Indian Institute of Technology Hyderabad, India*

**Rishabh Karnad** *ai22mtech12001@iith.ac.in*
*Department of Artificial Intelligence, Indian Institute of Technology Hyderabad, India*

**Mohammad Emtiyaz Khan** *emtiyaz.khan@riken.jp*
*RIKEN Center for AI Project, Tokyo, Japan*

**P.K. Srijith** *srijith@iith.ac.in*
*Department of Artificial Intelligence, Indian Institute of Technology Hyderabad, India*

## Abstract

Causal discovery aims to understand the relationships between individual random variables. In many applications, such as brain imaging and climate modeling, it is more meaningful to consider interactions among groups of variables. Existing methods assume that knowledge of such groups or clusters is explicitly available when modeling interactions. However, in practice, these clusters as well as the causal relationships among them, are latent. In this paper, we present a novel approach based on variational inference to simultaneously infer both the latent clusters and causal structures. We learn an approximate posterior over clusters and graph-structure by considering variational distributions based on categorical and Bernoulli models respectively. We derive variational lower bounds and estimation techniques to learn variational and model parameters. The effectiveness of our proposed methods for cluster and causal discovery are demonstrated on both synthetic and real data sets.

## 1 Introduction

The ability to go beyond statistical associations and infer causal effect relationships between variables in a data-generating system is crucial in many scientific applications (Pearl, 2009; Spirtes et al., 2000). Causal discovery aims to learn causal structure over a set of random variables from observational data, often represented as a directed acyclic graph (DAG). Typically, causal learning methods assume that the data generating process can be modeled using a structural causal model (SCM) (Peters et al., 2017). SCMs consist of a set of equations that represent independent mechanisms each of which produces scalar values and assumes that exogenous noise variables are independent. However, there are situations where it is more suitable to model mechanisms that generate vector-valued outputs. For instance, learning causal dependencies between brain regions rather than between individual neurons is more informative for neuroscientists (Semedo et al., 2020; Perich and Rajan, 2020).

In literature, DAGs over clusters, groups or partitions of variables are known as Group DAGs (Parviainen and Kaski, 2017) or Cluster DAGs (C-DAGs) (Anand et al., 2023). Wahl et al. (2023) discuss how to do causal inference over a pair of clusters of variables when the clustering is known. Another instance where grouping of variables is useful is in the presence of latent confounding, where the assumption of independent noise is violated. One of the works that group variables in the presence of confounding between exogenous variables is Kawahara et al. (2010), which explicitly assumes a linear non-Gaussian model. Vector-valued

SCMs (vSCMs) have been proposed recently by Wahl et al. (2024), which generalizes classical SCMs to accommodate grouping assumptions defined by C-DAG. Most works on C-DAGs assume variable grouping is known through domain knowledge. Recently, Niu et al. (2022) proposed a score-based approach to learn both clusters and DAGs jointly from the data, particularly when the underlying distribution is not faithful to the full DAG.

In this paper, we address the problem of jointly inferring causal structure and clusters from the observational data. We consider linear and non-linear Gaussian additive noise models based on vSCMs and propose a novel approach that treats both the clusters and structures as latent variables. The proposed approach learns an approximate posterior distribution over them by using variational inference (Blei et al., 2017). The proposed Bayesian treatment of learning distribution over clusters and structures can benefit in uncertainty quantification and active interventions. Bayesian learning has been found useful in the standard SCM setup in Annadani et al. (2021); Lorch et al. (2021); Deleu et al. (2022). However, developing a variational inference technique to infer both clusters and structures in vSCMs is challenging and we propose appropriate variational distributions and training methods to address this. The experimental results on synthetic and real data sets demonstrate the effectiveness of the proposed approach to jointly discover cluster and causal structure over clusters.

## 2 Background

### 2.1 Problem Statement

Let $\mathbf{X}$ be a $d$-dimensional random vector with elements $X_i \in \mathcal{R}$. Let $D$ be the dataset with $N$ observations of $\mathbf{X}$. Let $I$ be the index set with indices $\{1, \ldots, d\}$. We define a partitioning of the indices as $C = (C_1 ... C_k)$, denoting the $k$ clusters, where each cluster is a disjoint subset. Let $\mathbf{X}_{C_i}$ denote a vector containing all the values of $\mathbf{X}$ associated with index $C_i$. Let $k_i$ denote the number of elements in $\mathbf{X}_{C_i}$. A cluster-DAG or C-DAG is defined as the directed acyclic graph over the clusters in $C$, with an adjacency matrix denoted by $E_C$. An example of a C-DAG is given in Figure 1. We consider the problem of cluster DAG discovery where we need to infer the clusters and connections in the C-DAG $G_C = (C, E_C)$ from the observational samples $D$.

### 2.2 Vector-Valued Structural Causal Models

A vector-valued structural causal model (vSCM) (Wahl et al., 2024) is a set of vector-valued assignments defined over these clusters:

$$\mathbf{X}_{C_i} := f_{C_i}(\mathrm{Pa}(\mathbf{X}_{C_i}), \epsilon_i) \tag{1}$$

where $\mathrm{Pa}(\mathbf{X}_{C_i})$ is a vector containing a subset of $\mathbf{X}$ forming the parents of $\mathbf{X}_{C_i}$ and $\epsilon_i \in \mathbb{R}^{k_i}$ is a noise vector whose elements are possibly dependent. We denote the number of elements in $\mathrm{Pa}(\mathbf{X}_{C_i})$ as $\bar{k}_i$. The function $f_{C_i} : \mathbb{R}^{\bar{k}_i} \to \mathbb{R}^{k_i}$ is therefore a vector-valued function that defines the functional relationship between the vector $\mathbf{X}_{C_i}$ and its parent variables. The vSCM entails a graphical structure, which is a cluster DAG. The vSCM induces a joint distribution over the observed variables $p(\mathbf{X}|C, E_C; \Theta)$, where $\Theta$ are parameters of the distribution, that factorizes according to $C$ and $E_C$ as follows:

$$p\left(\mathbf{X}|C, E_C; \Theta\right) = \prod_{i=1}^{k} p\left(\mathbf{X}_{C_i}|\mathrm{Pa}(\mathbf{X}_{C_i}); \Theta_{C_i}\right) \tag{2}$$

where $\Theta_{C_i}$ are the local parameters of the conditional distribution over $\mathbf{X}_{C_i}$, obtained from the global parameter $\Theta$ using index $C_i$.

#### 2.2.1 Linear Gaussian Vector Valued SCMs

A linear-Gaussian vSCM considers the following form between variables and their parents:

$$\mathbf{X}_{C_i} = \Theta_{C_i}^T \mathrm{Pa}(\mathbf{X}_{C_i}) + \epsilon_i \tag{3}$$

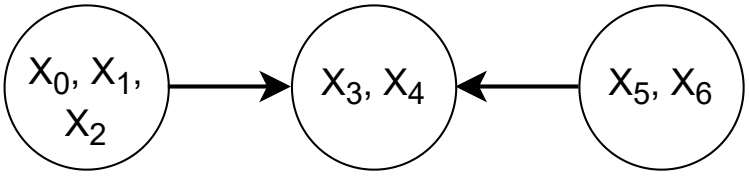



Figure 1: Example of a cluster-DAG

where $\Theta_{C_i}$ is a matrix representing a linear mapping from $\text{Pa}(\mathbf{X}_{C_i})$ to $\mathbf{X}_{C_i}$ and $\epsilon_i$ has a multivariate Gaussian distribution with zero mean and covariance $\Sigma_{C_i}$. The distribution of $\mathbf{X}$ therefore decomposes into the product of Gaussians,

$$P\left(\mathbf{X}|E_C, C, \Theta\right) = \prod_{i=1}^{k} \mathcal{N}\left(\mathbf{X}_{C_i}|\text{Pa}(\mathbf{X}_{C_i}); \mu_{C_i}, \Sigma_{C_i}\right) \tag{4}$$

where $\mu_{C_i} = \Theta_{C_i}^T Pa(\mathbf{X}_{C_i})$, and $\Theta$ is the collection of $\Theta_{C_1}...\Theta_{C_k}$.

### 2.2.2 Non-Linear Additive Noise Model

We also consider the non-linear additive noise model, where we model the $f_{C_i}$ in (1) as a neural network with $\Theta_{C_i}$ as weights associated to the mapping between $\text{Pa}(\mathbf{X}_{C_i})$ and $\mathbf{X}_{C_i}$. We model all $f_{C_i}$ with a single fully connected neural network with parameters $\Theta$ by masking the inputs and outputs based on $C_i$ and $Pa(C_i)$(Lorch et al., 2021; Zheng et al., 2020). Therefore the distribution of $\mathbf{X}$ can be written as:

$$P(\mathbf{X}|E_C, C; \Theta) = \prod_{i=1}^{k} \mathcal{N}(\mathbf{X}_{C_i}|\mu_{C_i}, \Sigma_{C_i}) \tag{5}$$

where $\mu_{C_i} = NN(Pa(X_{C_i}); \Theta_{C_i})$ is modelled as a neural network (NN).

## 2.3 Marginal Likelihood of DAG

For the case of standard DAG $G$, where the causal mechanism is linear Gaussian, the marginal likelihood $P(D|G)$ can be computed in closed form (Geiger and Heckerman, 1999) and is given as

$$P(D|G) = \prod_{i=1}^{d} \frac{P\left(D^{(X_i, \text{Pa}_G(X_i))}\right)}{P\left(D^{\text{Pa}_G(X_i)}\right)} \tag{6}$$

where $\text{Pa}_G(X_i)$ are the parents of the node $X_i$ in DAG $G$. Consider $\mathbf{Y}$ to be any subset of $\mathbf{X}$, i.e., $(X_i, \text{Pa}_G\left(X_i\right))$ or $(\text{Pa}_G\left(X_i\right))$, then $D^{\mathbf{Y}}$ is the dataset $D$ restricted to observations of $\mathbf{Y}$. For Gaussian DAG models with mean $\mu$ and precision matrix $W$ Geiger and Heckerman (1999) considered Normal-Wishart distribution as parameter priors. Then the parameters of Gaussian DAG model are marginalized out to give closed form expression for $D^{\mathbf{Y}}$ as given in (Geiger and Heckerman, 1999): $W \sim \mathcal{W}(T^{-1}, \alpha_w)$ and $\mu|W \sim \mathcal{N}(\nu, \alpha_\mu W)$, where $\nu$ is the prior mean, $T$ is the scale matrix, $\alpha_w > d-1$ is the degrees of freedom of the Wishart distribution and $\alpha_\mu > 0$ is a scaling parameter.

$$\begin{aligned} p(D^{\mathbf{Y}}) &= \frac{1}{\pi^{lN/2}} \left(\frac{\alpha_\mu}{N+\alpha_\mu}\right)^{l/2} \frac{\Gamma_l\left((N+\alpha_w-d+l)/2\right)}{\Gamma_l\left((\alpha_w-d+l)/2\right)} \frac{|T_{\mathbf{YY}}|^{(\alpha_w-d+l)/2}}{|R_{\mathbf{YY}}|^{(N+\alpha_w-d+l)/2}} \\ R &= T + S_N + \frac{N\alpha_\mu}{(N+\alpha_\mu)}(\nu-\bar{\mathbf{X}})(\nu-\bar{\mathbf{X}})^T \end{aligned} \tag{7}$$

where $l$ is the dimension of $\mathbf{Y}$. $\Gamma_l(.)$ is the multivariate Gamma function of dimension $l$. $T_{\mathbf{YY}}$ and $R_{\mathbf{YY}}$ are the sub-matrices of $T$ and $R$ respectively, containing the rows and columns corresponding to the elements in $\mathbf{Y}$. $\bar{\mathbf{X}}$ and $S_N$ are the empirical mean and $(N-1)\times(N-1)$ empirical covariance of $\mathbf{X}$ respectively.

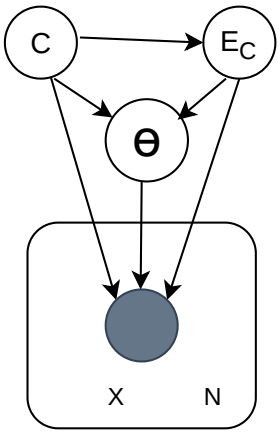



Figure 2: Graphical model of the generative process

## 3 Methodology

We propose a novel methodology to address the problem of jointly discovering clusters and their corresponding causal structures. We provide a principled approach based on Bayesian learning and inference to jointly infer them. In particular, we rely on variational inference to learn an approximate posterior distribution over the latent clusters and structures from the observational data. In a finite sample setting, several C-DAGs can be consistent with the observed data and learning a distribution over the cluster DAGs (C-DAGs) help us to discover all of them. It also allows us to quantify uncertainty over C-DAGs which can be crucial for several high-risk applications. Further, it allows one to do active interventions to obtain a better understanding of the underlying causal model.

### 3.1 Generative Model

To formalize our approach, we begin by specifying the generative process underlying the observed data. The graphical model that defines the generative process is given in Figure 2. We assume that the samples of $\mathbf{X}$ in $D$ are generated by a linear Gaussian vSCM as in (3) or non-linear additive model as in (5). In addition, $(C, E_C)$ are discrete *latent* random variables, where $C$ specifies the cluster assignments that form the nodes in the C-DAG and $E_C$ its edges. To maintain acyclicity, we assume $E_C$ to be a strictly upper triangular matrix that represents the adjacency matrix over those cluster nodes. Then the joint distribution $P(C, E_C, D, \Theta)$ is given by:

$$P(C, E_C, D, \Theta) = P(D|C, E_C, \Theta)\, P(\Theta|C, E_C)\, P(E_C|C)\, P(C)$$

where $P(C), P(E_C|C), P(\Theta|C, E_C)$ are the prior distributions and $P(D|\Theta, C, E_C)$ is the likelihood of modeling data $D$ given model parameters $\Theta$, and latent $C$, and $E_C$. We would like to infer $C$, and $E_C$ by learning the posterior distribution over them following Bayesian principles.

### 3.2 Prior Distributions

The Bayesian framework allows us to naturally encode domain knowledge about structures through priors. In addition, it plays a key role in estimating posterior over structures using Bayesian inference.

**Prior over clusters** In our work, we assume a uniform prior over clusters. With $d$ variables and $k$ clusters, the total number of possible cluster assignments is $k^d$. The prior distribution is given by $P(C) = \frac{1}{k^d}$.

**Prior over graph structure** In the space of DAGs, the number of denser graphs is greater than the number of sparser graphs (Eggeling et al., 2019). Consequently, fully connected graphs have a higher probability

mass. To balance out this effect, we choose a prior over $E_C$, which enforces sparsity and is given by:

$$P\left(E_C|C\right) \propto \exp\left(-\lambda_s \left\|E_C\right\|_1\right) \tag{8}$$

where $\lambda_s$ is the tunable hyperparameter. Unlike causal discovery algorithms (Annadani et al., 2021; Lorch et al., 2021), where the introduction of the acyclicity constraint in prior is essential to restrict the search space to DAGs, our parameterization of $E_C$ as an upper triangular matrix naturally enforces this constraint.

We discuss the posterior computation over $C$ and $E_C$ using Bayesian inference, for the linear and non-linear cases, separately in the following sections.

### 3.3 Linear Gaussian Vector Valued SCMs

#### 3.3.1 Marginal Likelihood

If the data is generated using linear Gaussian vSCM, the distribution of $\mathbf{X}$ is multivariate Gaussian. Thus, the choice of prior over parameters $\Theta$ and the assumptions such as likelihood modularity, prior modularity, and global parameter independence used to derive the marginal likelihood of DAG ($p(D|G)$) in 6 holds in this case. Consequently, the marginal likelihood $P(D|C, E_C)$ can be computed in closed form by integrating out $\Theta$. Since, for C-DAGs, the DAG is over the clusters, we obtain marginal likelihood as:

$$P(D|C, E_C) = \prod_{i=1}^{k} \frac{P\left(D^{\mathbf{X}_{C_i}, \mathrm{Pa}(\mathbf{X}_{C_i})}\right)}{P\left(D^{\mathrm{Pa}(\mathbf{X}_{C_i})}\right)} \tag{9}$$

where $\mathrm{Pa}(X_{C_i})$ are the parents of the vector $X_{C_i}$ in C-DAG $(C, E_C)$. The expression for each of the factors is obtained by using (7).

#### 3.3.2 Bayesian Inference over C-DAGs

By combining the prior and likelihood through Bayes rule, the posterior distribution over clusters and structures in C-DAGs is given as

$$\begin{aligned} P\left(C, E_C|D\right) =& \frac{P\left(D|C, E_C\right) P\left(C\right) P\left(E_C|C\right)}{P\left(D\right)} \\ =& \frac{P\left(D|C, E_C\right) P\left(C\right) P\left(E_C|C\right)}{\sum_{E_C} \sum_{C} P\left(D|C, E_C\right) P\left(C\right) P\left(E_C|C\right)} \end{aligned} \tag{10}$$

where $P\left(D\right)$ is the model evidence. Computing the posterior distribution in (10) in closed form requires us to compute summations over all C-DAGs in the evidence term. Since, the space of $C$ and $E_C$ over the $d$ variables and $k$ clusters grows in the order $\mathcal{O}\left(k^d\right)$ and $\mathcal{O}\left(k^2\right)$, respectively, this makes the computation infeasible. Hence, we will approximate the posterior $P\left(E_C, C|D\right)$ using variational distribution over $C$ and $E_C$ following the variational inference technique.

#### 3.3.3 Variational Inference for C-DAGs

Our goal is to learn the posterior distribution $P(C, E_C|D)$ given the observed dataset $D$. The variational inference framework casts the problem of inference as an optimization problem where the true posterior is approximated using the tractable family of distributions $Q_\phi\left(C, E_C\right)$ whose parameters are learned by minimizing the KL divergence between the approximate posterior and the true posterior. We consider the following factorization of the variational distribution over $C$ and $E_C$, parameterized by $\mathbf{T}$ and $\mathbf{Z}$.

$$P\left(C, E_C|D\right) \approx Q_\phi\left(C, E_C|D\right) = Q_{\mathbf{T}}\left(C\right) Q_{\mathbf{Z}}\left(E_C\right) \tag{11}$$

A direct minimization of the KL divergence requires computing the exact posterior. Variational Inference (Blei et al., 2017) provides an elegant way to compute the variational distribution by maximizing the *Evidence Lower Bound* ($ELBO$) instead, as in the following proposition.

**Proposition 1.** Let $Q_{\mathbf{T}}(C)$ and $Q_{\mathbf{Z}}(E_C)$ be the variational distribution of $C$ and $E_C$ respectively. Then the *evidence lower bound* ($ELBO$) is given by:

$$\begin{aligned}&\log P(D) \geq \mathcal{L}(\mathbf{Z}, \mathbf{T}) \\ &= \mathbb{E}_{Q_{\mathbf{Z}}(E_C)} \mathbb{E}_{Q_{\mathbf{T}}(C)} \left[ \log \frac{P(D|E_C, C) P(E_C, C)}{Q_{\mathbf{Z}}(E_C) Q_{\mathbf{T}}(C)} \right]\end{aligned}$$

### 3.4 Non-linear Vector Valued SCM

#### 3.4.1 Marginal Likelihood

For linear vSCMs, we obtained the marginal likelihood by marginalizing out $\Theta$ parameters of the causal model:

$$P(D|\, C, E_C) = \int_{\Theta} P(D|\Theta, C, E_C) P(\Theta|C, E_C)\, d\theta \tag{12}$$

However, such a marginalization is intractable, and not computable in closed form in the case of non-Linear vSCMs. So instead, we consider the following joint distribution and perform a point estimation of $\Theta$:

$$P(C, E_C, D|\Theta) = P(D|C, E_C; \Theta) P(E_C|C) P(C)$$

#### 3.4.2 Variational Inference for Non-linear vSCM

We can write the posterior $P(C, E_C|D; \Theta)$ as:

$$P(C, E_C|D; \Theta) = \frac{P(D|C, E_C; \Theta) P(C, E_C)}{P(D; \Theta)}$$

Due to the intractability in computing the marginal $P(D|\Theta)$, we resort to variational inference to obtain the posterior approximation. We consider the same variational approximation as in (11). However, the expression of the Evidence Lower Bound (ELBO), is different in this case and depends on the model parameters $\Theta$.

**Proposition 2.** Let $Q_{\mathbf{T}}(C)$ and $Q_{\mathbf{Z}}(E_C)$ be the variational distribution of $C$ and $E_C$ respectively, and $\Theta$ be the parameters of the neural network. Then the *evidence lower bound* ($ELBO$) is given by:

$$\begin{aligned}&\log P(D; \Theta) \geq \mathcal{L}(\mathbf{Z}, \mathbf{T}, \Theta) \\ &= \mathbb{E}_{Q_{\mathbf{Z}}(E_C)} \mathbb{E}_{Q_{\mathbf{T}}(C)} \left[ \log \frac{P(D|E_C, C; \Theta) P(E_C, C)}{Q_{\mathbf{Z}}(E_C) Q_{\mathbf{T}}(C)} \right]\end{aligned}$$

### 3.5 Variational Families

Our goal is to approximate an intractable joint distribution with a variational family that is tractable as well as flexible to model complex distributions over C-DAGs. We will first discuss the choice of variational family for $C$, then discuss the choice for $E_C$.

#### 3.5.1 Distribution over Clusters

In our work, we considered two models for $C$ a) Factorized model and b) Linear autoregressive model.

In **Factorized model**, we assume that each of the dimensions in $C$ is independent and the variational posterior of $C$ can be written as the product of independent categorical random variables given by:

$$\begin{aligned}Q_{\mathbf{T}}(C) &= \prod_{m=1}^{d} Q_{\mathbf{t_m}}(C_m) \\ Q_{\mathbf{t_m}}(C_m = i) &= \frac{\exp(t_{mi})}{\sum_{n=1}^{k} \exp(t_{mn})}\end{aligned} \tag{13}$$

where $i = \{1, \ldots, k\}, \mathbf{T} \in \mathbb{R}^{d \times k}$ is a matrix of logits.

The factorized model is a simpler model; such an approach may not be sufficient to capture the correlated density. There is inherent dependence among the dimensions that allows them to group together, which motivates us to look for a more expressive distribution. One natural way to capture this dependence is to learn the autoregressive distribution on $C$.

In **linear autoregressive model**, we model the variational posterior of $C$ as a product of conditionals, and each conditional is modeled with a categorical distribution given by:

$$Q_{\mathbf{T}}(C) = \prod_{m=1}^{d} Q_{\mathbf{t_m}}(C_m | C_{1:m-1}) \tag{14}$$

$$Q_{\mathbf{t_m}}(C_m = i | C_{1:m-1}) = \frac{\exp(t_{mi})}{\sum_{n=1}^{k} \exp(t_{mn})} \tag{15}$$

where $i = \{1, \ldots, k\}, \mathbf{T} \in \mathbb{R}^{d \times k}$ is a matrix of logits and $\mathbf{t_m}$ are the parameters of each conditional that are estimated using the linear autoregressive model. The linear autoregressive model is given by the following equation:

$$\begin{aligned} h_m &= Ah_{m-1} + b_{hh} + BC_{m-1} + b_{ch} \\ t_m &= Dh_m + c_{ho} \end{aligned} \tag{16}$$

where $C_{m-1}, h_m, t_m$ are the input, hidden state, and logits for $m^{th}$ variable. $A, B, D$ and $b$'s are parameters of the model.

### 3.5.2 Distribution over graphs

In our work, we considered four models for the adjacency matrix $E_C$: a) Independent binary model, b) Linear autoregressive model, c) Conditional model, and d) Autoregressive conditional model. Since $E_C$ is upper triangular, the dimension of $E_C$ is $k \times (k-1)/2$.

In the **independent binary model**, we assume that each entry in $E_C$ is an independent binary random variable. The variational distribution over $E_C$ is given as:

$$Q_{\mathbf{Z}}(E_C) = \prod_{i=1}^{k \times (k-1)/2} Q_{z_i}(e_{c_i}) = \prod_{i=1}^{k \times (k-1)/2} \frac{1}{1 + \exp(-z_i)} \tag{17}$$

where $\mathbf{Z} \in \mathbb{R}^{k \times (k-1)/2}$ is the vector of logits. The factorized model assumes independent edges, which may not be sufficient to capture the structural dependencies within the graph. To address this, we introduce three more expressive variational families for $E_C$.

In **linear autoregressive model** for graphs, we capture the topological dependence among the edges by modeling the variational posterior of $E_C$ as a product of conditionals, given by:

$$Q_{\mathbf{Z}}(E_C) = \prod_{i=1}^{k \times (k-1)/2} Q_{z_i}\left(e_{c_i} | e_{c_{1:i-1}}\right) \tag{18}$$

$$Q_{z_i}\left(e_{c_i} = 1 | e_{c_{1:i-1}}\right) = \frac{1}{1 + \exp(-z_i)} \tag{19}$$

where $z_i$ are the logits estimated using the linear autoregressive model:

$$\begin{aligned} h_i &= Ah_{i-1} + b_h + Be_{c_{i-1}} \\ z_i &= Dh_i + c_{he} \end{aligned} \tag{20}$$

**Algorithm 1** C-DAG Learning Using VI

**Input:** Dataset D
Initialize parameters $\mathbf{Z}, \mathbf{T}$
**while** not converged **do**
    $(C^i, E_C^i)_{i=1}^M \sim Q_{\mathbf{T}}(C), Q_{\mathbf{Z}}(E_C)$
    $G_{\mathbf{Z}} = \nabla_{\mathbf{Z}} \mathcal{L}(\mathbf{Z}, \mathbf{T})$ with $(C^i, E_C^i)_{i=1}^M$ ▷ use eq(25)
    $G_{\mathbf{T}} = \nabla_{\mathbf{T}} \mathcal{L}(\mathbf{Z}, \mathbf{T})$ with $(C^i, E_C^i)_{i=1}^M$ ▷ use eq(26)
    $\mathbf{Z} \leftarrow$ Update parameters using $G_{\mathbf{Z}}$
    $\mathbf{T} \leftarrow$ Update parameters using $G_{\mathbf{T}}$
**end while**

where $e_{c_{i-1}}, h_i, z_i$ are the input, hidden state, and logit for the $i^{th}$ edge. $A, B, D, b_h$, and $c_{he}$ are parameters of the model.

However, the variational families discussed earlier fail to accommodate the dependence of graph $(E_C)$ on the cluster assignments $(C)$. In the **conditional model**, we explicitly model the dependence of the edges on the cluster assignments $C$:

$$Q_{\mathbf{Z}}(E_C|C) = \prod_{i=1}^{k\times(k-1)/2} Q_{z_i}(e_{c_i}|C) \tag{21}$$

Here, the vector of logits $\mathbf{Z} \in \mathbb{R}^{k\times(k-1)/2}$ is defined as a linear transformation of the flattened one-hot representation of the cluster assignments, denoted as $C_{\text{flat}} \in \{0,1\}^{dk}$:

$$\mathbf{Z} = W_{E_C} C_{\text{flat}} + b_{E_C} \tag{22}$$

where $W_{E_C} \in \mathbb{R}^{k(k-1)/2\times dk}$ and $b_{E_C}$ are the learned weight matrix and bias parameters, respectively.

Finally, to simultaneously capture both the internal graph topology and its dependence on the global cluster assignments, we combine these approaches into an **autoregressive conditional model**. The variational posterior integrates both the previous edge states and the cluster assignments:

$$Q_{\mathbf{Z}}(E_C|C) = \prod_{i=1}^{k\times(k-1)/2} Q_{z_i}\left(e_{c_i}|e_{c_{1:i-1}}, C\right) \tag{23}$$

The logits $z_i$ are computed by incorporating both the autoregressive hidden state $h_i$ and the global cluster context $C_{\text{flat}}$:

$$\begin{aligned} h_i &= Ah_{i-1} + b_h + Be_{c_{i-1}} \\ z_i &= Dh_i + W_C C_{\text{flat}} + c_{bias} \end{aligned} \tag{24}$$

where $W_C$ projects the global cluster representation into the logit space for the $i^{th}$ edge alongside the autoregressive representation.

### 3.6 Gradient Approximation of ELBO

As described previously, in order to approximate the posterior over $(C, E_C)$ using variational inference, we estimate the parameters of the variational posterior by maximizing *ELBO*. We optimize *ELBO* using a gradient-based approach. This requires us to estimate the gradients $\nabla_{(\mathbf{Z},\mathbf{T})}\mathcal{L}(\mathbf{Z}, \mathbf{T})$. Since $C, E_C$ are discrete variables, we approximate the gradients using the score function estimator, REINFORCE (Williams, 1992). REINFORCE estimators are known to suffer from variance issues(Liévin et al., 2020), and in our work, we used an exponential moving average as the baseline $b$ similar to that used in Annadani et al. (2021) for variance reduction. Following this, the gradients are computed as:

$$\begin{aligned} \nabla_{\mathbf{Z}}\mathcal{L}(\mathbf{Z}, \mathbf{T}) = &\mathbb{E}_{Q_{\mathbf{Z}}(E_C)}\mathbb{E}_{Q_{\mathbf{T}}(C)}\Big[\nabla_{\mathbf{Z}} \log Q_{\mathbf{Z}}(E_C) \\ &\times \Big(\log \frac{P(D|E_C, C)\, P(E_C, C)}{Q_{\mathbf{Z}}(E_C)\, Q_{\mathbf{T}}(C)} - b\Big)\Big] \end{aligned} \tag{25}$$

**Algorithm 2** Non-Linear C-DAG Learning

**Input:** Dataset D
Initialize parameters $\mathbf{Z}, \mathbf{T}, \Theta$
**while** not converged **do**
  $(C^i, E_C^i)_{i=1}^M \sim Q_{\mathbf{T}}(C), Q_{\mathbf{Z}}(E_C)$
  $G_{\mathbf{Z}} = \nabla_{\mathbf{Z}} \mathcal{L}(\mathbf{Z}, \mathbf{T}, \Theta)$ with $(C^i, E_C^i)_{i=1}^M$ ▷ use eq(27)
  $G_{\mathbf{T}} = \nabla_{\mathbf{T}} \mathcal{L}(\mathbf{Z}, \mathbf{T}, \Theta)$ with $(C^i, E_C^i)_{i=1}^M$ ▷ use eq(28)
  $G_{\mathbf{\Theta}} = \nabla_{\mathbf{\Theta}} \mathcal{L}(\mathbf{Z}, \mathbf{T}, \Theta)$ with $(C^i, E_C^i)_{i=1}^M$ ▷ use eq(29)
  $\mathbf{Z} \leftarrow$ Update parameters using $G_{\mathbf{Z}}$
  $\mathbf{T} \leftarrow$ Update parameters using $G_{\mathbf{T}}$
  $\mathbf{\Theta} \leftarrow$ Update parameters using $G_{\mathbf{\Theta}}$
**end while**

$$\begin{aligned}\nabla_{\mathbf{T}} \mathcal{L}(\mathbf{Z}, \mathbf{T}) =& \mathbb{E}_{Q_{\mathbf{Z}}(E_C)} \mathbb{E}_{Q_{\mathbf{T}}(C)} \Big[ \nabla_{\mathbf{T}} \log Q_{\mathbf{T}}(C) \\ & \times \Big( \log \frac{P(D|E_C, C) P(E_C, C)}{Q_{\mathbf{Z}}(E_C) Q_{\mathbf{T}}(C)} - b \Big) \Big]\end{aligned} \tag{26}$$

The expectations in the gradient computations are approximated using Monte Carlo sampling. A summary of the proposed methodology is given in Algorithm 1.

For the non-linear formulation, we explore several variational families: for the clusters $C$, we utilize the factorized model (13) and the linear autoregressive model (16); for the graph $E_C$, we consider the independent binary model (17), the linear autoregressive model (20), the conditional model (22), and the autoregressive conditional model (24). Following Proposition 2, the gradients for variational parameters are computed as:

$$\begin{aligned}\nabla_{\mathbf{Z}} \mathcal{L}(\mathbf{Z}, \mathbf{T}, \Theta) =& \mathbb{E}_{Q_{\mathbf{Z}}(E_C)} \mathbb{E}_{Q_{\mathbf{T}}(C)} \Big[ \nabla_{\mathbf{Z}} \log Q_{\mathbf{Z}}(E_C) \\ & \times \Big( \log \frac{P(D|E_C, C; \Theta) P(E_C, C)}{Q_{\mathbf{Z}}(E_C) Q_{\mathbf{T}}(C)} - b \Big) \Big]\end{aligned} \tag{27}$$

$$\begin{aligned}\nabla_{\mathbf{T}} \mathcal{L}(\mathbf{Z}, \mathbf{T}, \Theta) =& \mathbb{E}_{Q_{\mathbf{Z}}(E_C)} \mathbb{E}_{Q_{\mathbf{T}}(C)} \Big[ \nabla_{\mathbf{T}} \log Q_{\mathbf{T}}(C) \\ & \times \Big( \log \frac{P(D|E_C, C; \Theta) P(E_C, C)}{Q_{\mathbf{Z}}(E_C) Q_{\mathbf{T}}(C)} - b \Big) \Big]\end{aligned} \tag{28}$$

As the distributions on $C$ and $E_C$ are independent of $\Theta$ in the ELBO, the gradient for $\Theta$ is computed as:

$$\begin{aligned}&\nabla_{\Theta} \mathcal{L}(\mathbf{Z}, \mathbf{T}, \Theta) \\ &\quad = \mathbb{E}_{Q_{\mathbf{Z}}(E_C)} \mathbb{E}_{Q_{\mathbf{T}}(C)} \left[ \nabla_{\Theta} \log P(D|E_C, C; \Theta) \right]\end{aligned} \tag{29}$$

All of the expectations are approximated with Monte Carlo sampling, the proposed methodology is summarized in Algorithm 2.

## 4 Experimental Results

In this section, we study the empirical performance of our method compared to the baseline (Niu et al., 2022) on the synthetic and real world datasets. Although most of the literature focuses on causal discovery for scalar variables, there are several works that study causal interaction on a group of variables (Wahl et al., 2024; Anand et al., 2023).

However, these works study theoretical aspects (Chalupka et al., 2016; Rubenstein et al., 2017) or the works in the direction of causal discovery assume that information about the groups is known (Wahl et al., 2023; Parviainen and Kaski, 2017). To the best of our knowledge, Niu et al. (2022) is the only work in the literature that focuses on the simultaneous estimation of clusters and graphs from observational data. This work gives the point estimates of the learned C-DAGs, however, we use bootstrapping to obtain multiple estimates

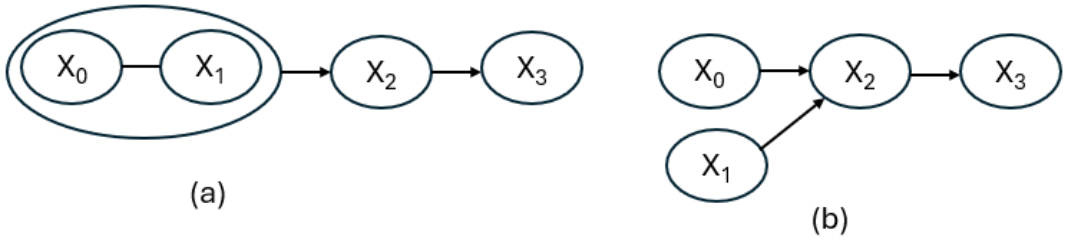



Figure 3: Example of a) C-DAG b) Expanded Graph

in order to compare the performance on multiple metrics. We use this as the baseline in our experiments. We will first discuss the evaluation criteria, then we will discuss the datasets considered in our work, the experimental setup and their corresponding results.

### 4.1 Evaluation Metrics

In case of smaller dimensions, i.e., $d <= 4$, enumeration of all DAGs is feasible and the exact posterior can be computed. In such scenarios, performance evaluation is done by computing the distance between the variational posterior and the ground truth posterior. Typically, **Hellinger distance (HD)** is used as a distance metric between distributions (Annadani et al., 2021). Compared to the space of all DAGs, the space of C-DAGs grows more slowly, with its complexity primarily determined by the number of clusters. Example for $k = 4$, one can enumerate C-DAGs upto $d \leq 7$. The other metrics used to evaluate the performance of causal discovery algorithms over DAGs, specifically for higher dimensions, are the structural Hamming distance (SHD) and the area under receiver operating curve (AUROC) (Annadani et al., 2021; Lorch et al., 2021). These metrics are intended to capture the similarity between the ground truth and the estimated graph structure. We take inspiration from the causal discovery literature and use the following metrics for evaluation.

**Expected Expanded-Graph SHD (ExG-SHD)** : Structural Hamming distance (Zheng et al., 2018) measures how many edges are different between two DAG structures. The **Expected SHD [E-SHD]** over the posterior is utilized in literature (Annadani et al., 2021; Lorch et al., 2021) to evaluate the performance of Bayesian causal discovery methods. We also compute the same metric on $E_C$ to evaluate the accuracy of the variational posterior.

However, this metric doesn't take into account the approximation error due to the variational posterior of $C$. In order to capture the impact of approximation error of both $C, E_C$ on the joint posterior, we define a new object called the **expanded-graph** and compute expected structural hamming distance on it. We call this metric **Expected Expanded Graph SHD (ExG-SHD)**. An example of a C-DAG and its corresponding expanded graph is shown in Figure 3. We define the expanded-graph as $G_{\text{expand}} = CE_CC^T$. and the ExG-SHD with respect to the ground truth expanded-graph $G^*$ is given by:

$$\begin{aligned} \mathbb{E}_{Q_{\mathbf{T}}(C)Q_{\mathbf{Z}}(E_C)}\left[\text{SHD}\right] &\approx \frac{1}{N}\sum_{i=1}^{N}\Big[\text{SHD}(G^{(i)}_{\text{expand}}, G^*)\Big] \\ C^{(i)}, E_C^{(i)} &\sim Q_{\mathbf{T}}(C), Q_{\mathbf{Z}}(E_C) \\ G^{(i)}_{\text{expand}} &= C^{(i)}E_C^{(i)}C^{(i)T} \end{aligned} \tag{30}$$

**Area Under Receiver Operating Curve (AUROC)** : Another commonly used metric in causal discovery literature (Annadani et al., 2021; Lorch et al., 2021) to evaluate the performance of Bayesian models is AUROC. We extend the metric to C-DAGs, where we compute edge beliefs over expanded graphs and compute the Receiver Operating Curve (ROC). The area under this curve is used as a metric to evaluate the performance.

Table 1: Results on linear synthetic datasets. ExG-SHD, E-SHD, and HD: lower is better; E-RI and AUC: higher is better. NA: Not Applicable.

(a) 4 variables, 3 clusters

| Data-sets | Methods | ExG-SHD | E-RI | AUC | E-SHD | HD |
|---|---|---|---|---|---|---|
| 4var 3clus fork | Baseline | 3.6 | 0.55 | 0.55 | 1.85 | NA |
| | Fact-C-E | 3.99 | 0.66 | 0.74 | 0.99 | 1.0 |
| | AR-C-Fact-E | **0.02** | 0.99 | **1.0** | **0.0** | **0.7** |
| | Fact-C-AR-E | 2.0 | **1.0** | 0.55 | 2.0 | 0.7 |
| 4var 3clus chain | Baseline | 3.6 | 0.55 | 0.42 | 1.9 | NA |
| | Fact-C-E | 1.0 | 0.66 | **0.79** | 0.0 | 1.0 |
| | AR-C-Fact-E | **0.03** | 0.67 | 0.6 | **0.0** | **0.71** |
| | Fact-C-AR-E | 2.00 | **1.00** | 0.58 | 2.0 | **0.71** |
| 4var 3clus vstruc | Baseline | 2.2 | 0.8 | 0.83 | 1.0 | NA |
| | Fact-C-E | 1.0 | 1.0 | 1.0 | 0.95 | 0.97 |
| | AR-C-Fact-E | **0.0** | **1.0** | **1.0** | **0.0** | **0.56** |
| | Fact-C-AR-E | 3.19 | 0.66 | 0.66 | 1.0 | 0.89 |

(b) 7 variables, 3 clusters

| Data-sets | Methods | ExG-SHD | E-RI | AUC | E-SHD | HD |
|---|---|---|---|---|---|---|
| 7var 3clus fork | Baseline | 10.8 | 0.3 | 0.55 | 1.9 | NA |
| | Fact-C-E | 10.0 | 1.0 | **0.77** | 2.0 | 0.71 |
| | AR-C-Fact-E | **4.0** | **1.0** | 0.74 | 2.0 | **0.71** |
| | Fact-C-AR-E | 13.9 | 0.65 | **0.81** | **1.0** | 1.0 |
| 7var 3clus chain | Baseline | 10.1 | 0.6 | 0.47 | 0.21 | NA |
| | Fact-C-E | 14.9 | 0.71 | 0.56 | 0.99 | 1.0 |
| | AR-C-Fact-E | 10.0 | 0.61 | **0.72** | **0.0** | **0.99** |
| | Fact-C-AR-E | **5.0** | **0.8** | **0.72** | 2.0 | 1.0 |
| 7var 3clus vstruc | Baseline | 10.95 | 0.2 | 0.68 | 1.2 | NA |
| | Fact-C-E | 11.9 | 0.71 | 0.8 | 1.0 | 1.0 |
| | AR-C-Fact-E | **6.8** | **0.75** | **0.95** | **1.0** | **0.99** |
| | Fact-C-AR-E | 10.3 | 0.63 | 0.75 | **1.0** | **0.99** |

**Expected Rand Index (E[RI])** : The similarity between the clusterings of the C-DAGs obtained from the estimated posterior are measured using rand-index (Hubert and Arabie, 1985). The expected rand-index with respect to the ground truth clustering $C^*$ is given by:

$$\begin{aligned} \mathbb{E}_{Q_{\mathbf{T}}(C)}\left[\mathrm{RI}\right] &\approx \frac{1}{N}\sum_{i=1}^{N}\Big[\mathrm{RI}(C^{(i)}, C^*)\Big] \\ C^{(i)} &\sim Q_{\mathbf{T}}(C) \end{aligned} \tag{31}$$

### 4.2 Experimental Setup

In our work, we conducted experiments on both synthetic and real-world datasets to evaluate how close our algorithm is able to approximate the posterior over $C, E_C$. In our experiments, we approximated $P\left(C|D\right)$ with both our linear and non-linear approaches and compared the results with respect to the baseline (Niu et al., 2022) for all datasets. Moreover, since the baseline is a non-Bayesian approach, to compare it with our Bayesian approach, we use bootstrapping (Friedman et al., 1999) to obtain multiple estimates by running the baseline on 20 bootstrapped datasets. For each run, we used 1000 iterations.

For both factorized and linear autoregressive models, we performed a hyperparameter tuning search for different learning rates $(0.1, 0.01, 0.001)$, optimizers (SGD, RMSprop, Adam) and learning rate schedulers (exponential, Cosine-Annealed LR). The RMSprop optimizer with a learning rate of 0.01 annealed by the exponential scheduler with gamma 0.9 at every 500 epoch; these settings were found to be optimal across all datasets. For the linear autoregressive model, we searched over the dimensions $(4, 8, 16, 32, 48, 64)$ of the hidden state (16). We chose the hyperparameters that result in maximum *ELBO*. Maximization of *ELBO* with respect to variational parameters is known to suffer from local optima, thus the solution heavily depends on the initial state. In our work, for each dataset and for each of the models, we do 10 random restarts. We run each of the experiments for $10,000$ epochs, with early stopping.

The Neural model was trained with the AdamW optimizer, with a cosine decay learning rate scheduler going from $5 \times 10^{-3}$ to $2 \times 10^{-5}$ for the first half of training. The 2 layer Neural Network hidden states were searched between $(16, 32, 64, 128)$. For the real world datasets, we do $75,000$ epochs, while for synthetic datasets we do $50,000$ epochs of training, and for all datasets and models we do 10 random restarts.

We consider multiple variants of the proposed model, differing in their variational families. Fact-C-E employs factorized models for both $C$ and $E_C$, while AR-C-Fact-E introduces an autoregressive model for $C$ alongside a factorized $E_C$. The remaining variants all maintain a factorized structure for $C$: Fact-C-AR-E applies an

Table 2: Results on nonlinear synthetic datasets. ExG-SHD and E-SHD: lower is better; E-RI and AUC: higher is better.

(a) 4 variables, 3 clusters

| Data-sets | Methods | ExG-SHD | E-RI | AUC | E-SHD |
|---|---|---|---|---|---|
| 4var fork | Baseline | 2.90 | **0.70** | 0.63 | 2.00 |
| | Fact-C-E | **0.00** | 0.67 | **1.00** | **1.00** |
| | AR-C-Fact-E | 4.88 | 0.59 | 0.87 | **1.00** |
| | Fact-C-AR-E | **0.00** | 0.67 | **1.00** | 2.00 |
| | Fact-C-Cond-E | **0.00** | 0.67 | **1.00** | **1.00** |
| | Fact-C-AR-Cond-E | **0.00** | 0.67 | **1.00** | 1.99 |
| 4var chain | Baseline | 2.80 | 0.70 | 0.62 | 2.00 |
| | Fact-C-E | 0.01 | **1.00** | **1.00** | **0.00** |
| | AR-C-Fact-E | 4.77 | 0.57 | 0.87 | 1.00 |
| | Fact-C-AR-E | 2.00 | **1.00** | 0.92 | 1.00 |
| | Fact-C-Cond-E | **0.00** | **1.00** | **1.00** | **0.00** |
| | Fact-C-AR-Cond-E | 2.00 | 0.99 | 0.87 | 1.00 |
| 4var vstruc | Baseline | 5.00 | 0.60 | 0.50 | 2.40 |
| | Fact-C-E | 1.00 | 0.67 | **1.00** | 1.01 |
| | AR-C-Fact-E | 5.27 | 0.61 | 0.94 | **1.00** |
| | Fact-C-AR-E | **0.01** | **0.83** | **1.00** | **1.00** |
| | Fact-C-Cond-E | 1.00 | 0.67 | **1.00** | **1.00** |
| | Fact-C-AR-Cond-E | 1.00 | 0.67 | 0.97 | **1.00** |

(b) 7 variables, 3 clusters

| Data-sets | Methods | ExG-SHD | E-RI | AUC | E-SHD |
|---|---|---|---|---|---|
| 7var fork | Baseline | 12.00 | 0.20 | 0.58 | 1.80 |
| | Fact-C-E | 6.00 | **0.81** | 0.99 | 1.00 |
| | AR-C-Fact-E | 17.05 | 0.57 | 0.48 | 1.00 |
| | Fact-C-AR-E | **0.01** | 0.71 | **1.00** | 1.68 |
| | Fact-C-Cond-E | **0.01** | 0.71 | **1.00** | **0.59** |
| | Fact-C-AR-Cond-E | **0.01** | 0.71 | **1.00** | 2.00 |
| 7var chain | Baseline | **11.70** | **0.80** | 0.74 | 1.90 |
| | Fact-C-E | 13.00 | 0.67 | 0.71 | **1.00** |
| | AR-C-Fact-E | 17.23 | 0.59 | 0.61 | **1.00** |
| | Fact-C-AR-E | 12.00 | 0.71 | 0.77 | **1.00** |
| | Fact-C-Cond-E | 12.00 | 0.71 | **0.78** | **1.00** |
| | Fact-C-AR-Cond-E | 12.00 | 0.71 | **0.78** | **1.00** |
| 7var vstruc | Baseline | 13.80 | 0.60 | 0.57 | 1.50 |
| | Fact-C-E | **0.00** | **0.81** | **1.00** | 0.57 |
| | AR-C-Fact-E | 17.81 | 0.58 | 0.59 | 1.00 |
| | Fact-C-AR-E | **0.00** | **0.81** | **1.00** | 0.07 |
| | Fact-C-Cond-E | **0.00** | **0.81** | **1.00** | **0.00** |
| | Fact-C-AR-Cond-E | **0.00** | **0.81** | **1.00** | **0.00** |

autoregressive model to $E_C$, Fact-C-Cond-E uses a conditional distribution for $E_C|C$, and Fact-C-AR-Cond-E leverages an autoregressive conditional architecture for $E_C|C$.

### 4.3 Synthetic Dataset

We generate samples for synthetic data using linear Gaussian vector-valued SCMs given by (4). Based on the number of clusters and fixed group sizes, we randomly generate the cluster assignments. To obtain the adjacency matrix over the clusters, we sample Erdos-Renyi graphs with edge probability 0.4 and ensure that the adjacency matrix generated is strictly upper triangular. Each of the elements of $\Theta \in \mathbb{R}^{d\times d}$ is sampled uniformly in $\{-2.,-5.\}\cup\{2.,5.\}$. The edge weights $\Theta_{C_i}$ are then obtained by masking $\Theta$ with the expanded graph. For the generation of synthetic data, we used a specific form of noise covariance $\Sigma_{C_i}$ that is added to each cluster. $\Sigma_{C_i}$ is given by:

$$\Sigma_{C_i} = \sigma\left(I_{k_i} + \rho\left(F_{k_i\times k_i} - I_{k_i}\right)\right)$$

where $I_{k_i} \in \mathbb{R}^{k_i\times k_i}$ is an identity matrix and $F_{k_i\times k_i} \in \mathbb{R}^{k_i\times k_i}$ is matrix of all 1's. $\rho$ allows control over the correlation between variables, and $\sigma$ adjusts the spread of the distribution. For all conditional distributions in (4), we used a covariance with the same $\sigma$ and $\rho$. Such a choice of sigma ensures that clusters in each group are correlated. For our experiments, we generated $N = 1000$ samples for $d = \{4, 7\}$, $k = 3$ with $\sigma = 0.1$ and $\rho = 0.9$. The fixed group sizes considered are $(2, 1, 1)$ and $(3, 2, 2)$ for $d = 4$ and $d = 7$ respectively. For 3 nodes, the interesting cases are chains, forks and v-structures, hence, we consider only those to generate our data. The C-DAGs considered are provided in the supplementary material. The non-linear dataset is generated in a similar manner, however a random non-linearity is applied to $\Theta_{C_i}^T \mathrm{Pa}(\mathbf{X}_{C_i})$ to generate $\mathbf{X}_{C_i}$. We generate $N = 3000$ samples for the non-linear data.

Since, the number of variables is small, enumeration of all CDAGs is possible, hence besides computing the metrics defined in the previous section, we were able to compute the distance between exact posterior and estimated posterior using Hellinger distance for the linear models. Baseline being a non-Bayesian approach, Hellinger distance computation is not applicable to the baseline.

The results for the synthetic datasets are provided in Tables 1a, 1b, 2a and 2b. Across all tested metrics on both the 4-variable and 7-variable synthetic datasets, we can see that our approach consistently performs better than the baseline method. In the case of linear formulations, we note that the autoregressive cluster

Table 3: Results on real-world datasets. NA means Not Applicable. ExG-SHD, E-SHD, HD lower is better. E-RI and AUC higher is better.

| Data-sets | Methods | ExG-SHD | E-RI | AUC | E-SHD | HD |
|---|---|---|---|---|---|---|
| Protein Dataset | Baseline | 3.0 | 0.55 | 1.0 | 0.3 | NA |
| | Lin Fact-C-E | 27.4 | 0.57 | 1.0 | 1.0 | 1.0 |
| | Lin AR-C-Fact-E | **0.0** | 0.56 | **1.0** | **0.0** | **0.7** |
| | Neur Fact-C-E | 0.04 | 0.74 | **1.0** | **0.0** | NA |
| | Neur AR-C-Fact-E | **0.0** | 0.56 | **1.0** | **0.0** | NA |
| | Neur Fact-C-Cond-E | **0.0** | **0.80** | **1.0** | **0.0** | NA |
| Climate Dataset | Baseline | 15.75 | 0.55 | 0.6 | 1.1 | NA |
| | Lin Fact-C-E | 20.1 | 0.55 | 0.31 | 0.0 | 1.0 |
| | Lin AR-C-Fact-E | 16 | **1.0** | 0.3 | 0.99 | **0.54** |
| | Neur Fact-C-E | 0.08 | **1.0** | **1.0** | **0.0** | NA |
| | Neur AR-C-Fact-E | 16 | 0.43 | 0.5 | 0.66 | NA |
| | Neur Fact-C-Cond-E | **0.04** | **1.0** | **1.0** | **0.0** | NA |

model generally outperforms the other variants, indicating the effectiveness of capturing the dependencies amongst the clusters. The baseline method struggles to accurately reconstruct the graphs, which is evident in its higher ExG-SHD scores, the metric that measures full graph recovery. In the case of linear formulations, the autoregressive cluster model generally outperforms the simpler factorized variant, indicating the practical necessity of capturing the dependencies amongst the clusters. For instance, in the 4-variable fork dataset (Table 1a), AR-C-Fact-E achieves an ExG-SHD of just 0.02 and an E-RI score of 0.99, tightly matching the ground truth.

In the nonlinear formulations, the performance gains are even more pronounced. The neural network is effectively able to learn the non-linearities allowing for our approach to vastly outperform than the baseline in most cases. As seen in Table 2a, several non-linear variants (such as Fact-C-E, Fact-C-AR-E, and Fact-C-Cond-E) achieve a flawless ExG-SHD of 0.00 for the 4-variable fork and v-structure configurations. Furthermore, we notice that both autoregressive and conditional models for edges outperform the factorized edge models, indicating that capturing topological edge dependencies yields better structural estimations than assuming the edges are entirely independent. Interestingly, the autoregressive cluster models do not perform as well in these settings.

### 4.4 Real World Datasets

In this section, we discuss the performance of our algorithm compared to the baseline on the two real-world datasets namely, Protein dataset and Climate dataset. The details of each of these datasets are provided in the appendix.

The results for the real-world datasets are provided in Table 3. Our approach performs better than the baseline in most of the metrics for real-world datasets. For both datasets, we were able to enumerate the exact posterior. The Hellinger distance indicates that the autoregressive model for clusters is able to approximate the true posterior better than the factorized cluster model.

In the case of the protein dataset, the ground-truth C-DAG does not have an edge, consisting instead of two separate, unconnected clusters. From Table 3, the metrics ExG-SHD and E-SHD that capture the precision of the structure are at the perfect values of 0 for the linear autoregressive cluster model, indicating its effectiveness on real-world datasets. In comparison, the baseline yields a much higher ExG-SHD of 3.0. The Neural models perform even better, with almost all of them achieving near perfect scores for ExG-SHD and E-SHD and demonstrating better clusterings with high E-RI scores. Neural Conditional Edge model in particular has the highest E-RI of 0.80, while sharing ExG-SHD and E-SHD scores of the linear autoregessive cluster model. This is in comparison to the baseline which yields an E-RI of 0.55.

The ground truth C-DAG for climate dataset has two clusters and an edge between. The neural models near perfectly recover the structure and clusters, indicating the non-linear nature of the underlying data. The

baseline model struggles here, resulting in a high ExG-SHD of 15.75 and an E-SHD of 1.1. In stark contrast, the neural formulations almost perfectly recover both the cluster assignments and the network structure. The Neural Conditional Edge model achieves a remarkably low ExG-SHD score of 0.04 and an E-SHD score of 0.0, accompanied with a perfect AUC of 1.0. It also demonstrates perfect E-RI score of 1.0, compared to the baseline score of 0.55. This exceptional recovery strongly suggests that the underlying causal mechanisms are inherently non-linear, and conditional modeling has notable improvement over the other methods.

## 5 Conclusion

We developed a variational inference technique for learning posterior over clusters and structures in C-DAGs, considering Bernoulli distribution over edges and categorical distribution over cluster assignments. Additionally, we examined two approaches for modeling cluster assignments: a factorized model and an autoregressive linear model to capture dependencies among assignments. We also examined multiple approaches to model the graph structure over these clusters, including an independent binary model, a linear autoregressive model, a conditional model, and an autoregressive conditional model.

To accommodate more complex data-generating processes, we successfully extended our framework to non-linear vector-valued SCMs by parameterizing the causal mechanisms with neural networks. The inference techniques were developed for both the linear and non linear models of vSCMs. The experiments on synthetic and real data sets showed the effectiveness of our approaches compared to the baselines. In the future, we would like to extend the work to scale it for high-dimensional datasets.

### References


T. V. Anand, A. H. Ribeiro, J. Tian, and E. Bareinboim. Causal effect identification in cluster dags. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12172–12179, 2023.

Y. Annadani, J. Rothfuss, A. Lacoste, N. Scherrer, A. Goyal, Y. Bengio, and S. Bauer. Variational causal networks: Approximate bayesian inference over causal structures. In *BCIRWIS 2021: Workshop on Bayesian Causal Inference for Real World Interactive Systems (KDD 2021 Workshop)*, 2021.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

K. Chalupka, F. Eberhardt, and P. Perona. Multi-level cause-effect systems. In *Artificial intelligence and statistics*, pages 361–369. PMLR, 2016.

T. Deleu, A. Góis, C. Emezue, M. Rankawat, S. Lacoste-Julien, S. Bauer, and Y. Bengio. Bayesian structure learning with generative flow networks. In *Uncertainty in Artificial Intelligence*, pages 518–528. PMLR, 2022.

R. Eggeling, J. Viinikka, A. Vuoksenmaa, and M. Koivisto. On structure priors for learning bayesian networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1687–1695. PMLR, 2019.

N. Friedman, M. Goldszmidt, and A. Wyner. Data analysis with bayesian networks: a bootstrap approach. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, page 196–205, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606149.

D. Geiger and D. Heckerman. Parameter priors for directed acyclic graphical models and the characterization of several probability distributions. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 216–225, Stockholm, Sweden, 1999. Morgan Kaufmann Publishers Inc.

L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

Y. Kawahara, K. Bollen, S. Shimizu, and T. Washio. Grouplingam: Linear non-gaussian acyclic models for sets of variables. *arXiv preprint arXiv:1006.5041*, 2010.

V. Liévin, A. Dittadi, A. Christensen, and O. Winther. Optimal variance control of the score-function gradient estimator for importance-weighted bounds. *Advances in Neural Information Processing Systems*, 33:16591–16602, 2020.

L. Lorch, J. Rothfuss, B. Schölkopf, and A. Krause. Dibs: Differentiable bayesian structure learning. *Advances in Neural Information Processing Systems*, 34:24111–24123, 2021.

X. Niu, X. Li, and P. Li. Learning cluster causal diagrams: An information-theoretic approach. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4871–4877, 7 2022.

P. Parviainen and S. Kaski. Learning structures of bayesian networks for variable groups. *International Journal of Approximate Reasoning*, 88:110–127, 2017.

J. Pearl. *Causality*. Cambridge university press, 2009.

M. G. Perich and K. Rajan. Rethinking brain-wide interactions through multi-region 'network of networks' models. *Current opinion in neurobiology*, 65:146–151, 2020.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

P. K. Rubenstein, S. Weichwald, S. Bongers, J. M. Mooij, D. Janzing, M. Grosse-Wentrup, and B. Schölkopf. Causal consistency of structural equation models. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence (UAI 2017)*, pages 808–817, 2017.

K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

J. D. Semedo, E. Gokcen, C. K. Machens, A. Kohn, and M. Y. Byron. Statistical methods for dissecting interactions between brain areas. *Current opinion in neurobiology*, 65:59–69, 2020.

P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, prediction, and search*. MIT press, 2000.

J. Wahl, U. Ninad, and J. Runge. Vector causal inference between two groups of variables. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12305–12312, 2023.

J. Wahl, U. Ninad, and J. Runge. Foundations of causal discovery on groups of variables. *Journal of Causal Inference*, 12(1):20230041, 2024. doi: 10.1515/jci-2023-0041.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

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

X. Zheng, C. Dan, B. Aragam, P. Ravikumar, and E. Xing. Learning sparse nonparametric dags. In *International conference on artificial intelligence and statistics*, pages 3414–3425. Pmlr, 2020.

## A Proof of Proposition 1

$$\begin{aligned}
&\arg\min_{\mathbf{Z},\mathbf{T}} KL(Q_{\mathbf{Z}}(E_C)\, Q_{\mathbf{T}}(C)\,||P(E_C, C|D)) \\
&= \mathbb{E}_{Q_{\mathbf{Z}}(E_C)}\mathbb{E}_{Q_{\mathbf{T}}(C)}\Big[\log Q_{\mathbf{Z}}(E_C) + \log Q_{\mathbf{T}}(C) - \log\left(\frac{P(D, E_C, C)}{P(D)}\right)\Big] \\
&= \mathbb{E}_{Q_{\mathbf{Z}}(E_C)}\mathbb{E}_{Q_{\mathbf{T}}(C)}\Big[\log Q_{\mathbf{Z}}(E_C) + \log Q_{\mathbf{T}}(C) - \log P(D, E_C, C) + \log P(D)\Big]
\end{aligned}$$

which gives us lower bound on the marginal log likelihood of the data

$$\log P(D) \geq \mathbb{E}_{Q_{\mathbf{Z}}(E_C)}\mathbb{E}_{Q_{\mathbf{T}}(C)}\Big[\log \frac{P(D|E_C, C)\, P(E_C, C)}{Q_{\mathbf{Z}}(E_C)\, Q_{\mathbf{T}}(C)}\Big]$$

## B Proof of Proposition 2

Consider the marginal log likelihood of the data

$$\begin{aligned}
\log P(D;\Theta) &= \log \sum_{E_C}\sum_{C} P(D|C, E_C;\Theta) P(E_C|C)\, P(C) \\
&= \log \sum_{E_C}\sum_{C} \frac{P(D|C, E_C;\Theta) P(E_C|C)\, P(C)\, Q_{\mathbf{Z}}(E_C)\, Q_{\mathbf{T}}(C)}{Q_{\mathbf{Z}}(E_C)\, Q_{\mathbf{T}}(C)}
\end{aligned}$$

By definition of Expectation:

$$= \log \mathbb{E}_{Q_{\mathbf{Z}}(E_C)}\mathbb{E}_{Q_{\mathbf{T}}(C)}\Big[\frac{P(D|C, E_C;\Theta) P(E_C|C)\, P(C)}{Q_{\mathbf{Z}}(E_C)\, Q_{\mathbf{T}}(C)}\Big]$$

By Jensen's inequality:

$$\geq \mathbb{E}_{Q_{\mathbf{Z}}(E_C)}\mathbb{E}_{Q_{\mathbf{T}}(C)}\Big[\log \frac{P(D|E_C, C;\Theta)\, P(E_C, C)}{Q_{\mathbf{Z}}(E_C)\, Q_{\mathbf{T}}(C)}\Big]$$

## C Data Generating C-DAGs for Real and Synthetic Datasets

In our work, we showed the performance of our algorithm on both synthetic and real-world datasets. Synthetic datasets are generated assuming a linear vector structural causal model. Synthetic data sets of $d = 4, 7$ and $k = 3$ were generated. For 3 nodes the interesting cases are chain, fork and v-structure, hence, we focused on these scenarios in our work.

Finding real datasets where information about DAGs over clusters is known is very difficult. In our work, we considered 2 real-world datasets Protein dataset and climate dataset. Protein dataset does not have any explicit information about ground-truth C-DAG. Looking at the DAG which is generated by experts, we can explicitly see two clusters which we considered as the ground truth. In Wahl et al. (2023) demonstrated their performance on the climate dataset. This motivates us to use this dataset. The ground truth CDAG used to generate data for each of the cases is given below:

### C.1 Data Generating C-DAG for 4 variable 3 cluster dataset

The Figure(4) shows data generating C-DAGs for $d = 4$ and $k = 3$.

### C.2 Data Generating C-DAG for 7 variable 3 cluster dataset

The Figure(5) shows data generating C-DAGs for $d = 7$ and $k = 3$.

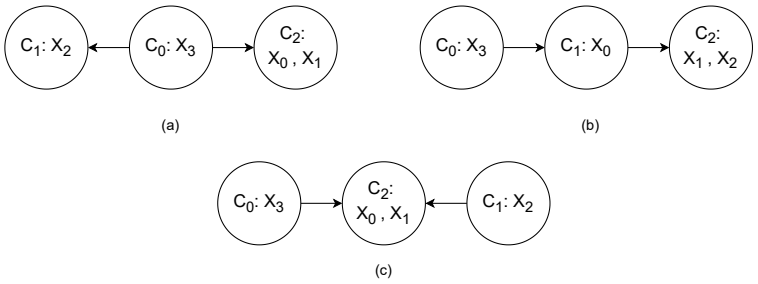



Figure 4: Data Generating CDAG for 4 variable 3 cluster synthetic dataset a) fork b) chain c) v-structure

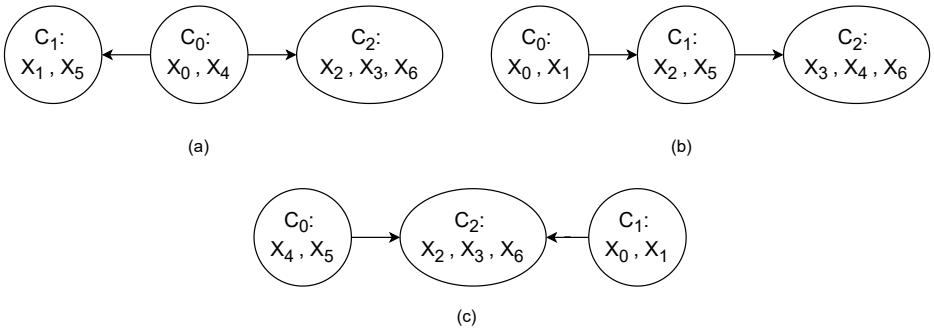



Figure 5: Data Generating CDAG for 7 variable 3 cluster synthetic dataset a) fork b) chain c) v-structure

### C.3 Non-Linearities for synthetic Non-Linear data

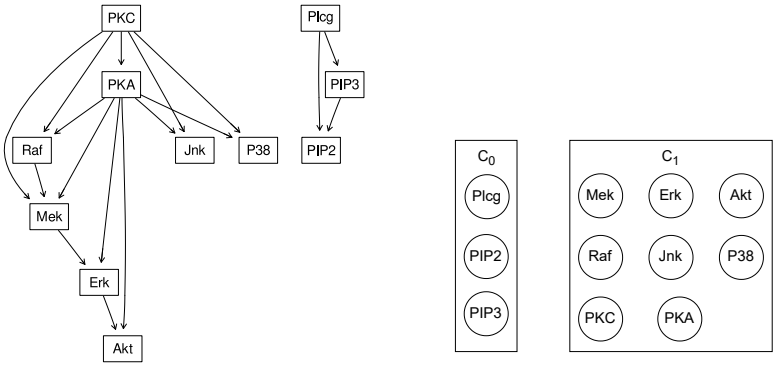
Our synthetic non-linear dataset is generated from the C-DAGs described in Figures 4 and 5. An element-wise non-linear operation is applied after the linear combination of parents to ensure non-linearity. Table 4 describes the used non-linearities.

Table 4: Element-wise non-linearities applied after linear combination of parents

| Dataset | | $C_1$ | $C_2$ |
|---|---|---|---|
| 4 variable 3 cluster | fork | Tanh | Tanh |
| | chain | Sin | ReLU |
| | v-structure | - | Cubic Polynomial |
| 7 variable 3 cluster | fork | Sin | ReLU |
| | chain | ReLU | Sin |
| | v-structure | - | Tanh |

### C.4 Details for Protein Dataset

We evaluated the performance of our algorithm on the protein signalling dataset (Sachs et al., 2005), which contains $d = 11$ features and $N = 853$ observations, with a ground truth graph provided by experts shown in Figure 6a. There is no true C-DAG known for this dataset. However, according to the ground truth structure, it is evident that there are two separate clusters which are not connected, i.e., one cluster containing nodes {Plcg, PIP3, PIP2} and the rest of the nodes in another cluster. The Figure(6a) shows data generating DAG given by experts. In our work we used Figure(6b) as the ground-truth C-DAG and is used for reporting metrics.




(a) Data-Generating DAG (b) Ground-truth C-DAG

Figure 6: Sachs Dataset

### C.5 Details for Climate dataset

We also evaluate our method on the climate dataset from (Wahl et al., 2023). The dataset consists of samples of surface temperatures in two geographical regions - the Tropical Pacific (ENSO) and British Columbia (BCT). The causal influence of temperature variations in the ENSO region on the BCT region is established in climate science, and the method described by (Wahl et al., 2023) recovers this relationship with around 59% accuracy. The dataset is generated by first deseasonalizing each sample by subtracting the monthly mean temperature of that geographical location, and Gaussian kernel smoothing with a bandwidth of $\sigma = 120$ months is used to remove long-term trends. Next, the temperatures from the months of October, November and December are averaged for the ENSO region, while those for the months of January, February and March are averaged for BCT in order to obtain 73 samples of annual temperatures, with BCT offset by one time lag. The resulting dataset contains 2 clusters, whose individual dimensionalities can be controlled by adjusting the size of grid boxes within each region. The climate data set consists of surface temperature samples from two regions, ENSO and BCT.

We considered surface temperature from four randomly chosen grid points in each region, hence $d = 8$ and $k = 4$. Figure(7) shows the ground-truth C-DAG for the climate dataset used to calculate the metrics.

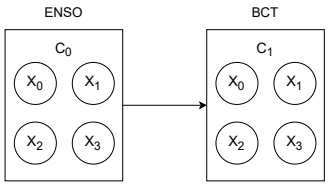



Figure 7: Ground truth C-DAG for climate dataset

## D Experimental Details

For both factorized and linear autoregressive models eq(16),figure(8), we performed a hyperparameter tuning search for different learning rates $(0.1, 0.01, 0.001)$, optimizers (SGD, RMSprop, Adam) and learning rate schedulers (exponential, Cosine-Annealed LR). The RMSprop optimizer with a learning rate of 0.01 annealed by the exponential scheduler with gamma 0.9 at every 500 epoch; these settings were found to be optimal across all datasets. We also varied $\lambda_s = (1, 10, 100)$. For 7 variable 3 cluster chain dataset $\lambda_s = 100$ was used, for remaining datasets $\lambda_s = 1$ was found to be suitable.

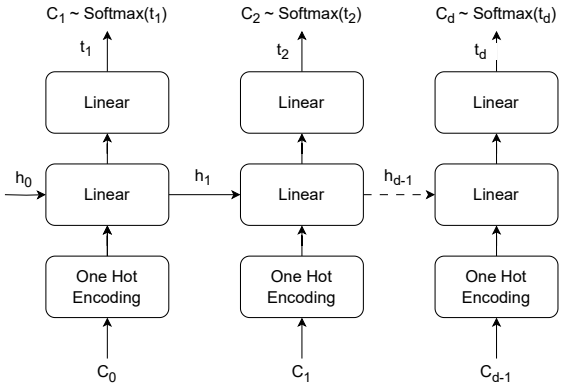



Figure 8: Autoregressive generation of cluster assignments using the linear model

In our work, for each dataset and for each of the models, we do 10 random restarts. We run each of the experiments for $10,000$ epochs. We used Xavier uniform for parameter initialization. All experiments are run on Tesla V100-SXM2-32GB GPU. For the linear autoregressive model, we searched over the dimensions $(4, 8, 16, 32, 48, 64, 128)$ of the hidden state. The final hidden state dimension that gave the best results is provided in Table(5). In addition, how much variation is across multiple restarts is shown Table(6).

For our neural formulation, we use a 2 hidden layer MLP with input and output layers being $d$ dimensional, and hidden layer activations being softplus. The hidden layer dimension of 64 for both layers was found to be optimal among $(16, 32, 64, 128)$. We use the AdamW optimizer to apply weight decay regularization for our neural parameters $\Theta$. $\lambda_s = 1000$ was found to be optimal for the protein signalling dataset, $\lambda_s = 30$ for the climate dataset, and $\lambda_s = 500$ across the synthetic nonlinear datasets.

Each experiment was run 10 times, and the learning was done for 75000 epochs for the real world datasets and for 50000 epochs for the synthetic datasets. A cosine decay scheduler for learning rate was used for the first half of training to lower the learning rate from $5 \times 10^{-3}$ to $2 \times 10^{-5}$. ELBO is approximated using 30 samples, and the performance metrics are calculated using 1000 samples from the predictive distribution.

Table 5: Linear Autoregressive Model hidden-state dimension choices

| Dataset | | Hidden-State Dimension |
|---|---|---|
| 4 variable 3 cluster | fork | 32 |
| | chain | 4 |
| | v-structure | 4 |
| 7 variable 3 cluster | fork | 8 |
| | chain | 128 |
| | v-structure | 48 |
| Sachs | | 4 |
| Climate | | 4 |

Table 6: Variation across random restarts for autoregressive linear cluster model for selected hidden dimension size

| Dataset | | E-RI±std | E-SHD±std |
|---|---|---|---|
| 4 variable 3 cluster | fork | 0.7±0.1 | 1.08±0.51 |
| | chain | 0.75±0.12 | 0.94±0.64 |
| | v-structure | 0.72±0.08 | 0.95±0.21 |
| 7 variable 3 cluster | fork | 0.68±0.07 | 1.05±0.22 |
| | chain | 0.60±0.06 | 0.97±0.5 |
| | v-structure | 0.65±0.06 | 1.0±0.0 |
| Sachs | | 0.51±0.051 | 0.57±0.49 |
| Climate | | 0.51±0.1 | 0.1±0.29 |