Differentiable Causal Discovery Under Latent Interventions

Gonçalo Rui Alves Faria
goncalorafaria@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2021

Abstract

Recent work has shown promising results in causal discovery by leveraging interventional data with gradient-based methods, even when the intervened variables are unknown. However, previous work assumes that the correspondence between samples and interventions is known, which is often unrealistic. We envision a scenario with an extensive dataset sampled from multiple intervention distributions and one observation distribution, but where we do not know which distribution originated each sample and how the intervention affected the system, *i.e.*, interventions are entirely latent. We propose a method based on neural networks and variational inference that addresses this scenario by framing it as learning a shared causal graph among a infinite mixture (under a Dirichlet process prior) of intervention structural causal models . Experiments with synthetic and real data show that our approach and its semi-supervised variant are able to discover causal relations in this challenging scenario.

Keywords: causal discovery, latent interventions, variational inference, Dirichlet process.

1. Introduction

Discovering causal relations among variables has countless applications in many scientific fields [25]. However, causal graphs are hard to learn from data; only with strong assumptions can we ensure that we will learn the correct causal structures from observations alone (even with an unlimited supply of data). Recent work by [4] has shown very promising results in causal discovery by leveraging interventional data with gradient-based methods, even when the intervened variables (targets) are not known. However, that work assumes that the *correspondence* between samples and interventions is known beforehand—often an unrealistic assumption.

Our work addresses this limitation/drawback by proposing a method for causal discovery under *fully latent* interventions, through a neural-based variational approach which infers the correspondences between samples and interventions from data. Our framework falls into the class of continuous constrained optimization methods for finding the DAG structure; other methods include constraint-based methods [31, 24, 32]. and score methods [6, 3, 10, 18]. We assume (as [4] do) that the data is clustered into intervention groups, but we relax their assumption that the correspondence between intervention groups and samples is known (see Figure 1), opening the door to more realistic scenarios. We model these *latent* interventions as being generated by a Dirichlet process prior and formulate the problem as one of end-to-end maximization of an evidence lower bound (ELBO). We summarize our contributions as follows:

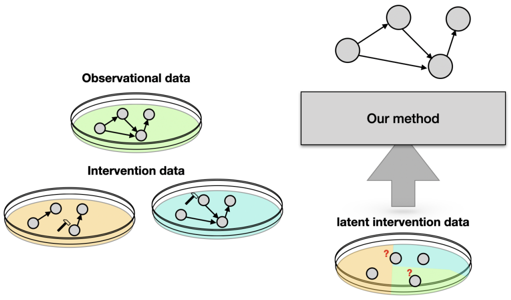

Figure 1: Illustrative example of the causal discovery scenario we consider.

- We formulate causal discovery under latent interventions as searching for the shared causal graph among an infinite mixture of intervened structural causal models.
- We propose a variational formulation with a Dirichlet process prior to model this infinite mixture. We use latent intervention embeddings with shared parameters to enable modeling an unlimited number of interventions.
- We develop a semi-supervised variant of our method; for when we know the correspondence for a subset of the samples.

Experiments on synthetic and real-world data show that our method is able to discover causal structures, outperforming several baselines and reaching similar performance to correspondence-aware methods.

2. Background

2.1. Structural causal models

We assume a structured causal model (SCM) $\mathcal{M} := (S, p(\mathcal{E}))$ over d *endogenous* variables $X = \{x_1, \dots, x_d\}$, associated to d independent *exogenous* variables $\mathcal{E} = \{\varepsilon_1, \dots, \varepsilon_d\}$, functions $\mathcal{F} = \{\zeta_1, \dots, \zeta_d\}$, and a collection of d assignments,

$$x_j := \zeta_j(\mathbf{PA}_j^{\mathcal{G}}, \varepsilon_j)\ , \qquad j = 1, \dots, d, \tag{1}$$

where $\mathbf{PA}_j^{\mathcal{G}} \subseteq X \backslash \{x_j\}$ is the set of *parents* of x_j according to a *directed acyclic graph* (DAG) $\mathcal{G}$, *i.e.*, $x_i \in \mathbf{PA}_j^{\mathcal{G}}$ if and only if there is a directed edge $x_i \to x_j$ in $\mathcal{G}$. An SCM $\mathcal{M}$ defines a unique joint distribution for X, usually referred to as the *entailed* distribution $p_{\mathcal{M}}(X)$, which can be factorized as

$$p_{\mathcal{M}}(X) = \prod_{j=1}^{d} p_{\mathcal{M}}(x_j | \mathbf{PA}_j^{\mathcal{G}}), \tag{2}$$

where $p_{\mathcal{M}}(x_j | \mathbf{PA}_j^{\mathcal{G}})$ is the conditional distribution of x_j, given its parents.

2.2. Interventions

Given an SCM $\mathcal{M}$, we obtain an *interventioned* SCM $\tilde{\mathcal{M}}$ by replacing one (or more) of the original assignments. Let I be the set of variables targeted by the intervention; if $I = \emptyset$, $\tilde{\mathcal{M}} = \mathcal{M}$. For each variable $j \in I$, the intervention consists in one or multiple of the following actions: replacing the assignment function ζ_j by $\tilde{\zeta}_j$; replacing the parents $\mathbf{PA}_j^{\mathcal{G}}$ by a subset $\tilde{\mathbf{PA}}_j^{\mathcal{G}}$; changing the noise variable from ϵ_j to $\tilde{\epsilon}_j$. The SCM $\tilde{\mathcal{M}}$ generally has a different entailed distribution, called the *intervention* distribution:

$$\begin{aligned} p_{\tilde{\mathcal{M}}}(X) = \prod_{j \notin I} p_{\mathcal{M}}(x_j | \mathbf{PA}_j^{\mathcal{G}}) \\ \prod_{j \in I} p_{\mathcal{M}; \mathrm{do}\left(x_j := \tilde{\zeta}_j(\tilde{\mathbf{PA}}_j^{\mathcal{G}}, \tilde{\varepsilon}_j)\right)}(x_j | \mathbf{PA}_j^{\mathcal{G}}). \end{aligned} \tag{3}$$

If there are K possible interventions, we denote the corresponding sets of target variables as $I^{(k)}$, for $k = 1, \dots, K$, and the corresponding SCMs by $\tilde{\mathcal{M}}^{(k)}$.

We divide the types of interventions into: *atomic*, if the target variable x_j is set to a constant value; *stochastic*, if x_j is set to a random variable $\tilde{\varepsilon}_j$; *imperfect* (or *soft*), if the intervention embedding and the set of parents are changed, as long as it does not become empty. We do not consider interventions that are able to add new elements to $\mathbf{PA}_j^{\mathcal{G}}$. This means that the intervention graph only differs from the observational by the removal of edges.

2.3. Faithfulness and Markov equivalence classes

Given a set $\mathcal{F}$ where each element ζ_i is *sufficiently* dependent on all of arguments $\mathbf{PA}_i^{\mathcal{G}}$, we obtain an SCM $\mathcal{M}$ whose computations strictly follow the structure of $\mathcal{G}$. In this scenario, $\mathcal{G}$ and $p_{\mathcal{M}}(X)$ are said to be mutually *faithful* since $\mathcal{G}$ encodes all and only the conditional independencies that hold in the entailed distribution. The set of faithful graphs that could entail a particular joint distribution is called the *Markov equivalence class* (MEC) [34]. If there is access to intervention data (in a set of interventions $\mathcal{I}$), it is possible to shrink the MEC to the so-called $\mathcal{I}$-MEC [9]: the subset of graphs in the MEC that have the same conditional independencies after applying the interventions in $\mathcal{I}$.

2.4. Continuous constrained optimization for structure learning

Our work builds on a recent line of research that uses continuous constrained optimization to address causal discovery, initiated by [36] and extended by [4] to cases where there is data from intervention distributions. In general, these methods adopt the *maximum a posteriori* (MAP) criterion (a.k.a. penalized maximum likelihood). Based on a generative/sampling model $p(\mathcal{D}|\mathcal{G}, \theta)$ for data $\mathcal{D}$, given the graph structure $\mathcal{G}$ and parameters θ, and on a *prior* $p(\mathcal{G})$ over graphs, they seek a graph that maximizes the score function

$$\mathcal{S}(\mathcal{G}) := \max_{\theta} \log p(\mathcal{D}|\mathcal{G}, \theta) + \log p(\mathcal{G}). \tag{4}$$

The prior $p(\mathcal{G})$ penalizes graph complexity to avoid over-fitting. A typical choice is $p(\mathcal{G}) \propto \exp(-\lambda|\mathcal{G}|)$, for $\lambda > 0$ and $|\mathcal{G}|$ is some graph complexity measure (*e.g.*, number of edges). With finite data, exact independence seldom occurs, thus graphs maximizing $\log p(\mathcal{D}|\mathcal{G}, \theta)$ alone would almost always be fully connected. If $\mathcal{D}$ is a collection of i.i.d. observations, then $p(\mathcal{D}|\mathcal{G}, \theta) = \prod_{i=1}^{n} p(x_i|\mathcal{G}, \theta)$.

Central to this class of methods is the weighted adjacency matrix $W^{\mathcal{G}} \in \mathbb{R}_{\geq 0}^{d \times d}$, where $W_{ij}^{\mathcal{G}} > 0$ is equivalent to $(i, j) \in \mathcal{G}$, which is treated as a parameter itself or as a function of the parameters. To ensure the estimated graph is a DAG, [36] proposed the constraint

$$\mathrm{trace}\left(e^{W^{\mathcal{G}}}\right) - d = 0, \tag{5}$$

where $e^{W^{\mathcal{G}}}$ is the matrix exponential. Several other methods apply non-linear models such as neural networks [19, 37] and define $W^{\mathcal{G}}$ differently.

The works from [23, 14, 4] treat the adjacency matrix as a random variable and relax the score from Eq. 4 in the following way:

$$\begin{aligned} &\mathcal{S}^{\star}(\Lambda) := \\ &\max_{\theta} \mathbb{E}_{\mathcal{G} \sim \mathrm{Bern}\left(\mathcal{G}; \sigma(\Lambda)\right)} \Big[\log p(\mathcal{D}|\mathcal{G}, \theta) + \log p(\mathcal{G}) \Big], \end{aligned} \tag{6}$$

where $\sigma(\Lambda)$ is the sigmoid transformation applied element-wise to the parameter matrix $\Lambda \in \mathbb{R}^{d \times d}$, $\mathrm{Bern}\big(\mathcal{G}; \sigma(\Lambda)\big)$ is a distribution over graphs, with mutually independent edges, with expected value $\sigma(\Lambda)$. This score tends asymptotically to $\mathcal{S}(\mathcal{G})$ as $\sigma(\Lambda)$ progressively concentrates its mass on a single DAG $\mathcal{G}$.

3. Differentiable causal discovery under latent interventions

In this section, we present a score for perfect or imperfect fully latent interventions, and show how this score can be approximately maximized by using an efficient variational optimization algorithm.

3.1. Mixture of intervention distributions

We assume that the dataset $\mathcal{D}$ is produced by a mixture of SCMs, each resulting from an intervention applied to a base SCM $\mathcal{M}$. More specifically, $\mathcal{D}$ is partitioned into $K+1$ exchangeable groups, with group k containing i.i.d. samples from the intervention SCM $\tilde{\mathcal{M}}^{(k)}$ resulting from applying the k^{th} intervention to the base SCM $\mathcal{M}$; the index $k=0$ indicates the absence of intervention, *i.e.*, $\tilde{\mathcal{M}}^{(0)} = \mathcal{M}$ (observational model). We denote by $\tilde{\mathcal{M}} = (\tilde{\mathcal{M}}^{(0)}, \dots, \tilde{\mathcal{M}}^{(K)})$ the ensemble of SCMs.

The latent variables $z^{(i)} \in \{0, \dots, K\}$ indicate which SCM generated each sample: $z^{(i)} = k$ if and only if $x^{(i)}$ is a sample of the SCM $\tilde{\mathcal{M}}^{(k)}$. Treating these correspondences $z^{(i)}$ as latent is a distinctive aspect of our work; while [4] also assume unknown $\tilde{\mathcal{M}}$, they assume that $z^{(i)}$ is observed, not latent. We call the scenario where both $\tilde{\mathcal{M}}$ and $z^{(i)}$ are unknown as *fully latent interventions.*

Marginalizing with respect to the latent $z^{(i)}$ yields the mixture model

$$\begin{aligned} p(x^{(i)}|\tilde{\mathcal{M}}) &= \sum_{k=0}^{K} p(z^{(i)} = k)\, p(x^{(i)}|z^{(i)} = k, \tilde{\mathcal{M}}) \\ &= \sum_{k=0}^{K} \tau_k\, p_{\tilde{\mathcal{M}}^{(k)}}(x^{(i)}), \end{aligned}$$

where we define $\tau_k = p(z^{(i)} = k)$. Conditioning on $z^{(i)}$ and invoking Equation 3 leads to

$$\begin{aligned} p(x^{(i)}|z^{(i)}, \tilde{\mathcal{M}}) = \sum_{k=0}^{K} \mathbb{I}(z^{(i)} = k) \prod_{j \notin I^{(k)}} p_{\mathcal{M}}(x_j^{(i)}|\mathbf{PA}_j^{\mathcal{G}}) \\ \prod_{j \in I^{(k)}} p_{\mathcal{M}; do\left(x_j := \tilde{\zeta}_j^k(\mathbf{PA}_j^{\mathcal{G}}, \tilde{\varepsilon}_j)\right)}(x_j^{(i)}|\mathbf{PA}_j^{\mathcal{G}}). \end{aligned}$$

We also consider that, like the group memberships, the set of targets $I^{(k)}$ of each intervention k is unknown (except for $I^{(0)} = \emptyset$).

3.2. Distribution over causal graphs

We represent the causal graph $\mathcal{G}$ via the adjacency matrix $A^{\mathcal{G}} \in \{0,1\}^{d \times d}$. Following previous work, our prior models each entry $A_{ij}^{\mathcal{G}}$, corresponding to edge $x_i \to x_j$, as a Bernoulli variable independent of all the others,

$$p(\mathcal{G}) = \prod_{i,j=1}^{d} \sigma(\lambda_{ij})^{A_{ij}^{\mathcal{G}}} \big(1 - \sigma(\lambda_{ij})\big)^{1 - A_{ij}^{\mathcal{G}}}, \qquad (7)$$

where $\sigma(u) = e^u/(1+e^u)$ is the usual logistic transformation (sigmoid) and the λ_{ij} are hyper-parameters. This prior over graphs is simplistic since it does not encode that $\mathcal{G}$ has to be a DAG. In this paper, we set $\lambda_{ij} = \lambda_{\mathcal{G}}$, for all i, j; however, in practice, a domain expert using the proposed method can embed prior knowledge in these hyper-parameters (our method can be straightforwardly adapted to that case).

The adoption of a probabilistic prior $p(\mathcal{G})$ should not be seen as expressing that it is in fact a random object, but rather as a subjective prior in the context of epistemic uncertainty about it.

3.3. Intervention embeddings and shared intervention space

We use density estimators, *e.g.*, neural networks and normalizing flows [27], to model the conditional densities in both the observational and interventional distributions. With this goal in mind, we use an appropriate encoding of the changes in the intervened assignments and the intervention targets. The set of targets in the k^{th} intervention $\tilde{\mathcal{M}}^{(k)}$ is indicated by a d-dimensional binary vector $r_k = [r_{k1}, \dots, r_{kd}]$, where $r_{kj} = 1$ if and only if $j \in I^{(k)}$, and $r_{kj} = 0$ otherwise. Since $I^{(0)}$ has no targets (it corresponds to the observational SCM $\mathcal{M}$), we have $r_0 = [0, 0, \dots, 0]$. To encode the type of intervention, we introduce the *intervention embedding vector* $u_k \in \mathbb{R}^h$, where h is an hyperparameter. The vector u_k represents the changes in the affected assignments for intervention $\tilde{\mathcal{M}}^{(k)}$. We denote by $\mathcal{R} = [r_0, r_1, \dots, r_K] \in \mathbb{R}^{K \times d}$ the matrix of intervention targets, and by $\mathcal{U} = [u_0, u_1, \dots, u_K] \in \mathbb{R}^{K \times h}$ the matrix of intervention embeddings. We use $(\mathcal{R}, \mathcal{U})$ as the representation of the interventions $\tilde{\mathcal{M}}$.

Putting everything together, when given the graph $\mathcal{G}$, interventions $\tilde{\mathcal{M}}$, indicator z, and assuming the intervention is imperfect, the log-probability of single datapoint x is given by

$$\begin{aligned} &\log p(x|z, \tilde{\mathcal{M}}, \mathcal{G}; \theta) = \qquad (8) \\ &= \sum_{j=1}^{d} \log g_j(x_j | A_j^{\mathcal{G}} \odot x, (e_z^\top \mathcal{R})_j\, (e_z^\top \mathcal{U} - u_0) + u_0; \theta_j), \end{aligned}$$

where $e_z \in \mathbb{R}^K$ is a one-hot vector indicating z, and $A_j^{\mathcal{G}} \odot x$ is the Hadamard (elementwise) product between the j^{th} column of the adjacency matrix of $\mathcal{G}$ and x, which is equivalent to selecting the entries of x in $\mathbf{PA}_j^{\mathcal{G}}$. The parameters $\theta = (\theta_1, \dots, \theta_d)$ parameterize the conditional densities $g_1, \dots, g_d$. We will consider in the sequel several forms for these conditional densities, *e.g.*, using parametric families and normalizing flows. Crucially, each conditional density g_j is a distribution of x_j, and the parameters θ_j are *shared* between all of the interventions—only the intervention-specific intervention embedding vector u_k changes depending on the intervention. This enables dealing with an unlimited number of interventions, as we shall see.

3.4. Modeling the conditional densities

A simple nonlinear model for the conditional densities g_j can be constructed using neural networks. We use a neural network $\text{NN}([u_k, A_j^{\mathcal{G}} \odot x]; \theta_j) : \mathbb{R}^{(h+d)} \rightarrow \mathbb{R}^m$, a parametric non-linear mapping, parameterized by θ_j, that receives the concatenation of the parents of x_j and the intervention embedding u_k and outputs m parameters of some distribution $f(x_j; \text{NN}([u_k, A_j^{\mathcal{G}} \odot x]; \theta_j))$ of the variable x_j. There are many possible choices for the distribution f, depending on the problem at hand and on whether x_j is discrete or continuous: Poisson ($m = 1$), Bernoulli ($m = 1$), univariate Gaussian ($m = 2$), categorical, etc. In this paper, we focus on three density families in $\mathbb{R}$, which we experiment with in Section 4.

Linear Gaussian We use a neural network $\text{NN}(u_k; \theta_j)$ to output coefficients $\tilde{a}_j \in \mathbb{R}^d$, and $\tilde{\sigma}_j, \tilde{b}_j \in \mathbb{R}$. Then, we use these as parameters of a Gaussian distribution whose mean is an affine transformation of the values of the parents of x_j:

$$g_j(x_j | A_j^{\mathcal{G}} \odot x, u_k) = \mathcal{N}\big(\tilde{a}_j^\top (A_j^{\mathcal{G}} \odot x) + \tilde{b}_j, \tilde{\sigma}_j^2\big).$$

Non-Linear Gaussian We use a neural network $\text{NN}([u_k, A_j^{\mathcal{G}} \odot x]; \theta_j)$ to output coefficients $\tilde{\mu}_j \in \mathbb{R}$ and $\tilde{\sigma}_j \in \mathbb{R}$, given the values of the parents of x_j as input. Then, we use these as parameters of a Gaussian distribution:

$$g_j(x_j | A_j^{\mathcal{G}} \odot x, u_k) = \mathcal{N}\big(\tilde{\mu}_j, \tilde{\sigma}_j^2\big).$$

Normalizing flows To model non-linear non-Gaussian conditional densities, we employ normalizing flows. A normalizing flow [27] is a transformation of a base probability density (in our case a Gaussian) through a sequence of invertible mappings $\tau(x_j; \tilde{W}_j) = \tau_l \circ \tau_{l-1} \cdots \circ \tau_1(x_j; \omega_1)$, where $\tilde{W}_j = \{\omega_1, \ldots, \omega_l\}$. We use a model introduced in [12], called Deep Sigmoidal Flows (DSF), where each of the invertible mappings has the following form:

$$\begin{aligned} &\tau_l(x) = \sigma^{-1}(w_l^\top \sigma(a_l x + b_l)), \\ &w_l \in \triangle_{F-1},\ a_l \in \mathbb{R}_+^F,\ b_l \in \mathbb{R}^F. \end{aligned}$$

where $\triangle_{F-1}$ is the probability simplex and F is an hyper-parameter. We use a neural network $\text{NN}([u_k, A_j^{\mathcal{G}} \odot x]; \theta_j)$ to output the parameters $\tilde{W}_j$. With the former, we obtain a controllable flow $\tau(x_j; \tilde{W}_j)$ that when given x_j, outputs the parameters of a Gaussian distribution $\tilde{\mu}$, $\tilde{\sigma}$. Altogether, the joint density has the following form:

$$g_j(x_j | A_j^{\mathcal{G}} \odot x, u_k) = \left| \det \left(\frac{\partial \tau(x_j; \tilde{W}_j)}{\partial x_j} \right) \right| \mathcal{N}\Big(\tilde{\mu}, \tilde{\sigma}^2\Big).$$

3.5. Modeling latent interventions with a Dirichlet process

To obtain a complete statistical description of the data-generating process, we still need to design a prior distribution for the latent interventions, apart from the sampling model g_j. Namely, we need a prior distribution for the correspondence z, the intervention embeddings $\mathcal{U}$, and the intervention targets $\mathcal{R}$. Furthermore, while experimentally, we can design scenarios where K, the number of latent interventions, is known, in general, given a data set, it will not be clear what the number of latent interventions is. Therefore, we will formulate the model to support a potentially non-specified number of latent interventions K.

We do this by using a *Dirichlet process prior* with a stick-breaking process representation [7, 30] as the prior distribution of the variables associated with the latent interventions. The generative story is as follows. We first draw the graph $\mathcal{G}$ from $p(\mathcal{G})$, as described in Equation 7. Then, for $k = 0, 1, \ldots,$ we sample the variables u_k and r_k, associated with each intervention $\tilde{\mathcal{M}}^{(k)}$, as well as the probability β_k of picking that intervention as a stick-breaking process, with scaling parameter $\alpha > 0$ (which controls the clustering effect of the Dirichlet process) and hyperparameter γ (which controls the sparsity of the intervention targets), as follows:

$$\begin{aligned} &u_k \sim \mathcal{N}(0, I_h) \\ &r_{kj} \sim \text{Bern}(\sigma(\gamma)), \quad j = 1, \ldots, d \\ &\beta_k = v_k \prod_{k'=0}^{k-1} (1 - v_{k'}), \quad \text{with } v_k \sim \text{Beta}(1, \alpha). \end{aligned}$$

Then, to generate the data, we first sample the intervention index $z^{(i)}$, and then sample a point $x^{(i)}$ conditioning on the corresponding intervention $\tilde{\mathcal{M}}^{(z^{(i)})}$:

$$\begin{aligned} &z^{(i)} \sim \text{Cat}(\beta_0, \beta_1, \ldots, \beta_k, \ldots) \\ &x_j^{(i)} \sim \begin{cases} g_j(x_j | A_j^{\mathcal{G}} \odot x^{(i)}, u_0) & \text{if } r_{z^{(i)}j} = 0 \\ g_j(x_j | A_j^{\mathcal{G}} \odot x^{(i)}, u_{z^{(i)}}) & \text{if } r_{z^{(i)}j} = 1. \end{cases} \end{aligned}$$

Figure 2 contains a graphical model representation of this joint distribution.

3.6. Variational approximation

To obtain the marginal likelihood $p(\mathcal{D}|\mathcal{G}; \theta)$, it is necessary to marginalize over the joint distribution of the model present in Section 3.5, *i.e.*:

$$\begin{aligned} p(\mathcal{D}|\mathcal{G}; \theta) &= \prod_{i=1}^{N} p(x^{(i)}|\mathcal{G}; \theta) \qquad (9) \\ &= \prod_{i=1}^{N} \mathbb{E}_{z^{(i)}, \tilde{\mathcal{M}} \sim p(z^{(i)}, \tilde{\mathcal{M}})} \big[p(x^{(i)}|z^{(i)}, \tilde{\mathcal{M}}, \mathcal{G}; \theta)\big]. \end{aligned}$$

However, the exact maximization of this marginal log-likelihood (which involves a product of Gaussians, Beta distributions, and complex conditional distributions

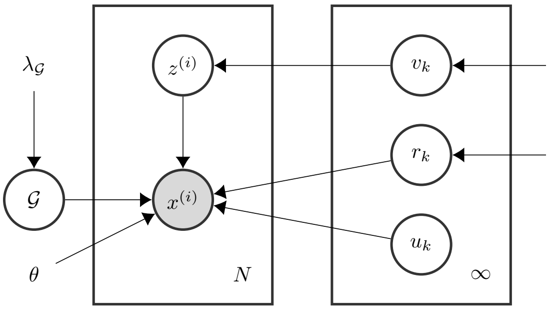

Figure 2: Graphical model representation of the Dirichlet Process Mixture model augmented for the causal discovery problem.

generated by neural networks) is intractable. Therefore, we resort to *approximate variational inference* [2]. We design a family of tractable variational distributions $q_\phi(z^{(i)}, \tilde{\mathcal{M}})$ to approximate the true posterior $p(z^{(i)}, \tilde{\mathcal{M}}|x^{(i)}, \mathcal{G}; \theta)$. For the variables associated with latent interventions $\tilde{\mathcal{M}}$ we propose the fully factorized and finite variational family

$$q(\tilde{\mathcal{M}}) = \prod_{k=0}^{K} \Big(\prod_{j=1}^{d} q_R(r_{kj})\Big)\Big(\prod_{l=1}^{h} q_U(u_{kl})\Big) q_V(v_k),$$

where here the hyper-parameter K defines the truncation level of the variational approximation, and the distributions associated with v_k, u_{kl}, and r_{kj} take the following form:

$$\begin{aligned} q_V(v_k; \rho_k, w_k) &= \text{Beta}\big(\rho_k w_k, (1-\rho_k) w_k\big); \\ q_U(u_{kl}; \mu_{kl}, \sigma_{kl}) &= \mathcal{N}(\mu_{kl}, \sigma_{kl}^2); \\ q_R(r_{kj}; \pi_{kj}) &= \text{Bern}(\pi_{kj}). \end{aligned}$$

where $\pi_{kj}, \mu_{kl}, \sigma_{kl}, \rho_k, w_k$ are free parameters to be optimized, for each $k \in \{0, \dots, K\}$, $l \in \{1, \dots, h\}$ and $j \in \{1, \dots, d\}$. For the distribution of interventional assignments z, we propose the following variational posterior:

$$q_Z(z) \propto \exp\left(\frac{u_z^\top \text{NN}(x; \phi_Z)}{\sqrt{h}}\right), \qquad k = 1, \dots, K\ ,$$

where $\text{NN}(x; \phi_Z) : \mathbb{R}^d \to \mathbb{R}^h$ is a neural network. We provide details in Appendix A about the derivation and, in the case of the beta distribution, closed-form approximation of the Kullback–Leibler divergence between the proposals and the prior. We use the shorthand ϕ to denote the vector of all variational parameters, which includes $\phi_Z, \mu_{kl}, \sigma_{kl}, \rho_k, w_k, \pi_{kj}$ for all $k \in \{0, \dots K\}$, $l \in \{1, \dots, h\}$, and $j \in \{1, \dots, d\}$.

Given the ingredients above, we obtain the following lower bound for the marginal likelihood from Equation 9:

$$\begin{aligned} &\log p(\mathcal{D}|\mathcal{G}; \theta) \geq \\ &\underbrace{\underbrace{\sum_{i=1}^{N} \mathbb{E}_{z^{(i)}, \tilde{\mathcal{M}} \sim q(z^{(i)}, \tilde{\mathcal{M}}; \phi)} \big[\log p(x^{(i)}|z^{(i)}, \tilde{\mathcal{M}}, \mathcal{G}; \theta)\big]}_{} \\ &-D_{KL}\big[q(z^{(i)}, \tilde{\mathcal{M}})||p(z^{(i)}, \tilde{\mathcal{M}})\big]}_{\text{ELBO}_{q(z^{(i)}, \tilde{\mathcal{M}}; \phi)}(x^{(i)}, \mathcal{G}; \theta)} \qquad (10) \end{aligned}$$

For different choices of ϕ, we get different lower bound approximations to the marginal likelihood. By maximizing the ELBO w.r.t. ϕ we minimize the approximation gap, which equals he KL divergence between the approximate and the true posterior.

3.7. A score for latent interventions

Using the log-likelihood proposed in Equation 9, we write a new score function $\mathcal{S}(\mathcal{G})$ for our model with latent interventions, with an associated relaxation to support a weighted adjacency $\sigma(\Lambda)$, as shown in Equation 6. As discussed in Section 3.6, in general we will not be able to exactly maximize this score. However, using the variational approximation from Equation 10, and associated variational family $\mathcal{Q}$, we can approximate the score $\mathcal{S}(\mathcal{G})$ with a surrogate score $\mathcal{S}_{\mathcal{Q}}(\mathcal{G}; \phi)$, for any ϕ, as follows:

$$\begin{aligned} &\mathcal{S}(\mathcal{G}) \geq \\ &\underbrace{\max_{\theta} \sum_{i=1}^{N} \text{ELBO}_{q(z^{(i)}, \tilde{\mathcal{M}}; \phi)}(x^{(i)}, \mathcal{G}; \theta) + \log p(\mathcal{G})}_{\mathcal{S}_{\mathcal{Q}}(\mathcal{G}; \phi)}\ . \end{aligned}$$

Observing equation

$$\begin{aligned} &\mathcal{S}_{\mathcal{Q}}(\mathcal{G}; \phi) = \\ &\mathcal{S}(\mathcal{G}) -\ D_{KL}\big[q(z^{(i)}, \tilde{\mathcal{M}}; \phi)||p(z^{(i)}, \tilde{\mathcal{M}}|x^{(i)}, \mathcal{G}; \theta^*)\big], \end{aligned}$$

we notice that, by maximizing the $\mathcal{S}_{\mathcal{Q}}(\mathcal{G}; \phi)$ w.r.t. ϕ we minimize the approximation gap between $\mathcal{S}(\mathcal{G})$ and $\mathcal{S}_{\mathcal{Q}}(\mathcal{G}; \phi)$, which equals the Kullback–Leibler divergence between the approximate and the true posterior. Given this insight, we propose the surrogate score $\mathcal{S}_{\mathcal{Q}}(\mathcal{G})$, where the following inequality holds for all ϕ:

$$\begin{aligned} &\mathcal{S}(\mathcal{G}) \geq \\ &\underbrace{\max_{\theta, \phi} \sum_{i=1}^{N} \text{ELBO}_{q(z^{(i)}, \tilde{\mathcal{M}}; \phi)}(x^{(i)}, \mathcal{G}; \theta) + \log p(\mathcal{G})}_{\mathcal{S}_{\mathcal{Q}}(\mathcal{G})} \\ &\geq \mathcal{S}_{\mathcal{Q}}(\mathcal{G}; \phi). \qquad (11) \end{aligned}$$

As many score-based causal discovery methods do, we can relax the surrogate score from Equation 11, yielding

$$\begin{aligned} \mathcal{S}^*_{\mathcal{Q}}(\Lambda) = \max_{\theta, \phi} \mathbb{E}_{\mathcal{G} \sim \text{Bern}\big(\mathcal{G}; \sigma(\Lambda)\big)} \Big[\qquad (12) \\ \sum_{i=1}^{N} \text{ELBO}_{q(z^{(i)}, \tilde{\mathcal{M}}; \phi)}(x^{(i)}, \mathcal{G}; \theta) + \log p(\mathcal{G})\Big]. \end{aligned}$$

The gap between the relaxed score $\mathcal{S}^*(\Lambda)$ and our surrogate $\mathcal{S}^*_{\mathcal{Q}}(\Lambda)$ tends asymptotically to the KL divergence between the best approximate posterior from $\mathcal{Q}$ and the true posterior, as $\sigma(\Lambda)$ progressively concentrates its mass on single DAG $\mathcal{G}$, more concretely:

$$\mathcal{S}^*(\Lambda) - \mathcal{S}^*_{\mathcal{Q}}(\Lambda) = \mathbb{E}_{\mathcal{G}\sim\text{Bern}\left(\mathcal{G};\sigma(\Lambda)\right)}\Big[D_{KL}\big[q(z^{(i)},\tilde{\mathcal{M}};\phi^*)||p(z^{(i)},\tilde{\mathcal{M}}|x^{(i)},\mathcal{G};\theta^*)\big]\Big],$$

where θ^* and ϕ^* are respectively the model and variational parameters that maximize Equation. 12.

3.8. Inference algorithm

The surrogate score, coupled with the acyclicity constraint from [36], enables us to formulate causal discovery under latent interventions as the following optimization problem:

$$\Lambda^* = \arg\max_{\Lambda} \mathcal{S}^{\mathcal{Q}}(\Lambda)$$
$$\text{s.t.} \underbrace{\text{Tr}\left(e^{\sigma(\Lambda)}\right) - d}_{h(\Lambda)} = 0, \qquad (13)$$

Following [36], we use the augmented Lagrangian procedure [11, 26, 8] to transform the problem in Equation 13 into a sequence of unconstrained optimization subproblems. When we estimate the gradients using the path-wise gradient estimator [17, 28], each unconstrained optimization subproblem reduces to sampling the graph and the latent variables from the variational posteriors using the reparametrization trick, minimizing the following objective:

$$\mathcal{L}(\theta,\phi,\Lambda,z,\tilde{\mathcal{M}},\mathcal{G};x,\mu_t,\varphi_t) = - \log p(x|z,\tilde{\mathcal{M}},\mathcal{G};\theta) + \Omega(\phi) - \lambda_{\mathcal{G}}||\Lambda||_1 + \varphi_t h(\Lambda) + \frac{\mu_t}{2}h(\Lambda)^2,$$

where $\Omega(\phi)$ denotes the sum of the Kullback–Leibler divergences, and μ_t and φ_t are the parameters of the augmented Lagrangian at the t^{th} iteration. For estimating the gradients of Λ, $q(z)$ and $q(r_{kj})$ we used a Gumbel-softmax continuous relaxation [13, 21] which, for the causal graph's distribution, was combined with the straight-through gradient estimator [1], to make sure the graph samples actually represented the hard dependencies of the SCM, instead of fractional ones. Having estimated the gradients, for each sample, we average them and feed them to the first order stochastic optimization algorithm Adam [15]. We implemented our method with the PyTorch framework [20].

3.9. Semi-supervised extensions

The causal discovery algorithm can be extended to cases where we have information about the correspondences $z^{(i)}$ and/or the intervention targets $I^{(k)}$. To achieve this, we only have to ignore the corresponding variational posteriors and use the observed values $z^{(i)}$ and $r^{(i)}$ as constants. We designate the original version of our model the **latent** variant, the one with observed $z^{(i)}$ the **unknown** variant, and the one with observed $z^{(i)}$ and $r^{(i)}$ the **known** variant. It is straightforward to extend our method to a semi-supervised setting—a scenario where we only observe the correspondence z for a fraction of the samples. Under this scenario, we still use the samples from the variational posterior to predict the unobserved values of z and use the observed ones to improve the variational posterior. In essence, with the semi-supervised variant, we obtain a model that interpolates between the latent variant and the unknown variant. We follow the approach from [16] and extend the ELBO objective (in our case, the lower bound of the distribution $p(\mathcal{D}|\mathcal{G};\theta)$). Let $\mathcal{O}$ be the set of the indices for which the intervention assignment z is known, and with $\hat{z}^{(i)}$ its particular value. We rewrite the bound on the log-likelihood as :

$$\text{log}p(\mathcal{D}|\mathcal{G};\theta) \geq \sum_{i=1}^{N} \text{ELBO}_{\theta,q(\hat{z}^{(i)},\tilde{\mathcal{M}})}(x^{(i)},\mathcal{G}) + \kappa\, \mathbb{I}(i\in\mathcal{O})\, \mathbb{E}_{\tilde{\mathcal{M}}\sim q(\tilde{\mathcal{M}})}\big[\log q(\hat{z}^{(i)}|x^{(i)},\tilde{\mathcal{M}};\phi_z)\big],$$

where in the first term, when $i \in \mathcal{O}$, we do not sample from $q(z^{(i)}|x^{(i)},\tilde{\mathcal{M}};\phi_z)$, using the observed value $\hat{z}^{(i)}$ instead. The variable $\kappa \in]0,1[$ is an hyper-parameter that controls the relative importance of supervised component of the objective. Using this bound, we can write a new relaxed surrogate score $\mathcal{S}^*_{\text{SLL}}(\Lambda)$ as:

$$\mathcal{S}^*_{\text{SSL}}(\Lambda) = \mathbb{E}_{\mathcal{G}\sim\text{Bern}\left(\mathcal{G};\sigma(\Lambda)\right)}\Big[\sum_{i=1}^{N} \text{ELBO}_{\theta,q(\hat{z}^{(i)},\tilde{\mathcal{M}})}(x^{(i)},\mathcal{G}) + \kappa\, \mathbb{I}(i\in\mathcal{O})\, \mathbb{E}_{\tilde{\mathcal{M}}\sim q(\tilde{\mathcal{M}})}\big[\log q(\hat{z}^{(i)}|x^{(i)},\tilde{\mathcal{M}};\phi_z)\big]\Big].$$

4. Experiments

We tested our method on synthetic and real data. The synthetic datasets allow us to do a systematic, controlled comparison of different methods in different scenarios (graph size and density, intervention and assignment types). In order to generate SCMs, perform interventions on them, and sample the corresponding entailed distributions, we created a Pytorch package for this purpose.[1]

Single-node interventions. We generated SCMs with $d = 10$ variables with a Erdős-Rényi scheme with expected number of edges per node $e \in \{1,4\}$. We generated 10 SCMs for each combination of e, conditional density (linear Gaussian, non-linear Gaussian, and normalizing flow), and intervention type (stochastic and

[1] The code will be provided upon acceptance.

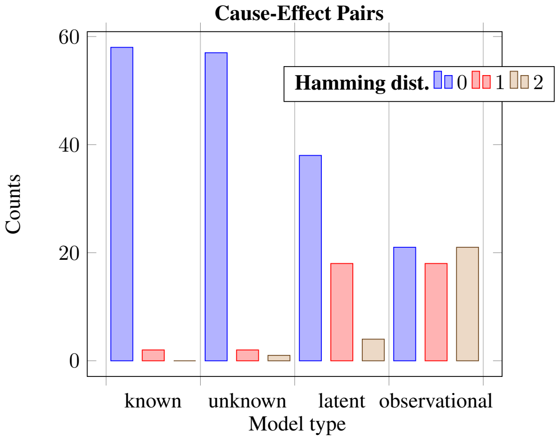

Figure 3: Histogram of Hamming distance in the experiments with cause-effect pairs.

Model	e	latent	unknown	known	observational
		Stochastic Interventions:			
L. G.		5.9 ± 6.2	3.4 ± 3.2	0.5 ± 1.3	10.3 ± 7.8
N. L. G.	1	12.2 ± 3.9	10.3 ± 2.5	7.0 ± 3.6	13.7 ± 3.8
N. F.		8.7 ± 6.6	8.0 ± 2.7	6.6 ± 2.2	11.3 ± 5.0
L. G.		27.2 ± 6.2	24.1 ± 5.8	15.6 ± 6.0	39.6 ± 5.0
N. L. G.	4	35.8 ± 3.8	30.3 ± 5.3	27.7 ± 4.3	37.5 ± 5.2
N. F.		36.1 ± 4.4	35.5 ± 8.1	31.5 ± 5.6	40.2 ± 6.9
		Imperfect Interventions:			
L. G.		5.8 ± 4.2	6.2 ± 3.06	4.7 ± 3.6	10.4 ± 2.9
N-L. G.	1	9.3 ± 2.4	8.9 ± 2.5	7.8 ± 3.9	10.5 ± 2.8
N. F.		8.8 ± 3.0	9.1 ± 3.5	7.9 ± 1.4	11.5 ± 5.4
L. G.		35.9 ± 8.3	29.7 ± 5.6	17.7 ± 7.9	39.1 ± 9.1
N. L. G.	4	32.1 ± 6.0	32.6 ± 5.8	32.8 ± 5.4	39.8 ± 9.3
N. F.		30.4 ± 12.2	30.2 ± 11.2	25.8 ± 3.9	36.7 ± 9.8

Table 1: Hamming distances on synthetic 10 variable SCMs. Our models L.G., N.L G., and N.F. stand for, respectively, linear Gaussian, non-linear Gaussian, and normalizing flows.

imperfect). In each of the SCMs, we performed 1 interventions for every variable. For each SCM within each configuration, we explored the following hyperparameter range: $\lambda_{\mathcal{G}} \in \{-.1, -.01, 0, .01, .1\}$, and $\gamma \in \{-.1, -.01\}$. For the remaining hyperparameters, we set $\alpha = 9$, $h = 248$ and the truncation level $K = 11$—the hyperparameter configuration that achieved the best log-likelihood on a validation set. From the generated SCM, we produced a dataset with $n = 10000$ samples, where each intervention has $\lfloor \frac{n}{d+1} \rfloor$ elements. This dataset was split into training (80%) and validation (20%). The models were trained for 500 epochs for the first iteration of the augmented Lagrangian and 50 epochs for the remaining ones, with a full batch($B = 8000$) until $h(\Lambda) < 10^{-8}$, and learning rate of $10^{-2.5}$. The results from these experiments are shown in Table 1. The metric we use is Hamming distance (HD) between the adjacency matrices of the ground-truth graph and the estimated one. The columns from Table 1 correspond to the different variants we present in Section 3.9: **latent** refers to our model, **unknown** assumes that the correspondences are known (as [4]), and **known** assumes that both correspondences and intervention targets are known. We additionally include a "naive" observational model (a baseline which ignores the existence of intervention data, *i.e.*, that assumes $K = 1$ as if all data was generated by the observational model). The experimental results preponderantly show that our method (the latent variant) consistently outperforms the observational baseline and is worst than the unknown variant, as expected, but only slightly. This indicates that taking account latent interventions, when these are present, improves the recovery of the causal graph.

Single-node interventions on cause-effect pairs. We generated two-variable linear Gaussian SCMs (cause-effect pairs) with edge probability $e = \frac{2}{3}$ (equal probability for each of the three possible graphs). The SCMs were generated as described in Appendix **??**. We sampled 60 SCMs, and generate a dataset of $n = 999$ samples, where each intervention has 333 elements. The sampled graph $\mathcal{G}$, with variables A and B is one of 3 possible graphs, A causes B, B causes A, or A and B have no cause-effect relation. All of the possible graphs share the MEC. We compare the variants presented in Section 3.9 in addition to the naive observational baseline. Figure 3 contains an histogram of Hamming distances for each of the variants. Edges in the anti-causal direction cost HD $= 2$, missing edge or wrong edge is HD $= 1$, and correct graph is HD $= 0$. The results show for this simple problem that cannot be identified using observational data alone, that our method correctly identifies a significant majority of the cause-effect pairs, even without information about the intervention assignments.

Real-world data set Finally, we tested our method on the flow cytometry data set of [29]. The measurements are the level of expression of phosphoproteins and phospholipids in humans cells. Interventions were performed by using reagents to activate or inhibit the measured proteins. The dataset comprises 7466 items with 11 variables each. Figure 4 contains an illustration of the consensus graph from [29]. This graph contains 11 edges. Table 2 compares the estimated graph, under different conditional density assumptions but assuming imperfect interventions, to the consensus graph from [29]. The results from the real-world data set show that our method outperforms several baselines, even with methods that use information regarding intervention correspondences and targets. The reasons that might justify relatively good results on this standard problem is that i) the causal sufficiency assumption may not hold, ii) the interventions may not be as specific as stated, and iii) the

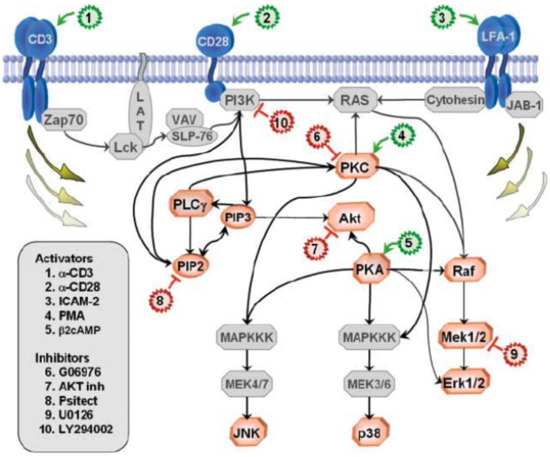

Figure 4: Classic signaling network and points of intervention. This is a graphical illustration of the conventionally accepted signaling molecule interactions, the events measured, and the points of intervention by small-molecule inhibitors. This illustration was obtained from [29].

	SHD	tp	fn	fp	rev	F_1 score
IGSP [35]	18	4	6	5	7	0.42
GIES [9]	38	10	0	41	7	0.33
CAM [5]	35	12	1	30	4	0.51
DCDI-G [4]	36	6	2	25	9	0.31
DCDI-DSF [4]	33	6	2	22	9	0.33
L. G. **(ours)**	33	7	11	22	3	0.30
N. L. G. **(ours)**	19	7	11	8	0	0.42
N. F. **(ours)**	30	9	9	21	1	0.38

Table 2: Results for the flow cytometry dataset. The results for the baselines are reproduced from [4]. Our models L.G., N.L G., and N.F. stand for, respectively, linear Gaussian, non-linear Gaussian, and normalizing flows. They all assumed imperfect interventions.

ground truth network is possibly not a DAG since feedback loops are common in cellular signaling networks as noted by [4]. These reasons can potentially be detrimental to the other methods and our method appears to be robust to them.

5. Related Work

The previous work that is closest to ours in that by [4]. As they do, we assume that the data is clustered in intervention groups, but we relax their assumption that the correspondence between intervention groups and samples is known, opening the door to more realistic scenarios.

The so-called "known" and "unknown" variants of our method share the assumptions made by [4]; however, contrary to them, the number of distinct neural networks in our model is not dependent on the number of interventions, which allows scaling well to many interventions. We achieve this by encoding the change in assignments associated with each intervention using a specific latent variable u that we call intervention embedding and conditioning a shared model on it when computing the log-probability.

Our method can be seen as a *variational autoencoder* (VAE) with a Dirichlet process (DP) prior, with a stick-breaking process representatoion. The work by [22] introduces a VAE where the stick-breaking weights are the latent variables. Our model differs from theirs in that we use the entire DP (including the atoms). We only use the stick-breaking weights in the KL divergence of the correspondence variable z. This KL divergence has a closed-form, so we do not sample the stick-breaking weights as they do. [33] proposes a VQ-VAE model with discrete latent variables, each represented as a latent embedding vector (an atom). Our approach is similar in that we use atoms (the intervention embeddings and intervention targets) to represent discrete latent variables. However, we do it in a statistically sounder way that more naturally fits our application.

6. Conclusion

We introduced an efficient variational optimization algorithm for causal structure learning under latent interventions. Our results are competitive with other state-of-the-art algorithms on the flow cytometry data set. Experiments with synthetic data show that our approach can recover causal relations even in our more challenging scenario, and it can consistently outperform our purely observational alternative.

A limitation of our method is the variational family we use. The proposal for variational posterior considers that the stick-breaking weights, the intervention embeddings, and intervention targets are independent of each other. In some cases, this can potentially create a surrogate score whose maximum structure is not an element of the $\mathcal{I}$-Markov equivalence class.

There are many avenues for future work. Our framework is particularly appealing for problems where performing interventions explicitly is expensive or unethical, but where interventions occur naturally in the data without being explicitly observed. Experimenting with more flexible variational families also seems appealing, albeit this may come at the cost of the closed-form expressions for the KL divergences.

Acknowledgements

I would like to express my appreciation to the supervisors, André Martins and Mário Figueiredo, for the valuable guidance, discussions, and support. I also would like to thank the DeepSPIN/MAIA research group for providing the computational resources that allowed me to adequately experiment with and evaluate the proposed methods.

References

[1] Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013.

[2] D. Blei, A. Kucukelbirb, and J. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[3] R. R. Bouckaert. Probabilistic network construction using the minimum description length principle. In M. Clarke, R. Kruse, and S. Moral, editors, *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 41–48, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[4] P. Brouillard, S. Lachapelle, A. Lacoste, S. L. Julien, and A. Drouin. Differentiable causal discovery from interventional data. *CoRR*, abs/2007.01754, 2020.

[5] P. Bühlmann, J. Peters, and J. Ernest. CAM: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526 – 2556, 2014.

[6] D. M. Chickering and D. Heckerman. Efficient approximations for the marginal likelihood of bayesian networks with hidden variables. In *Machine Learning*, pages 29–181, 1997.

[7] T. S. Ferguson. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1(2):209 – 230, 1973.

[8] R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. 1975.

[9] A. Hauser and P. Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *J. Mach. Learn. Res.*, 13(1):2409–2464, Aug. 2012.

[10] D. Heckerman, D. Geiger, and D. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. In R. L.de Mantaras and D. Poole, editors, *Uncertainty Proceedings 1994*, pages 293–301. Morgan Kaufmann, San Francisco (CA), 1994.

[11] M. R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

[12] C. W. Huang, D. Krueger, A. Lacoste, and A. C. Courville. Neural autoregressive flows. *CoRR*, abs/1804.00779, 2018.

[13] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax, 2017.

[14] D. Kalainathan, O. Goudet, I. Guyon, D. Lopez-Paz, and M. Sebag. Structural agnostic modeling: Adversarial learning of causal graphs, 2020.

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[16] D. P. Kingma, D. J. Rezende, S. M., and M. Welling. Semi-supervised learning with deep generative models. *CoRR*, abs/1406.5298, 2014.

[17] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2014.

[18] J. Kuipers, G. Moffa, and D. Heckerman. Addendum on the scoring of gaussian directed acyclic graphical models, 2021.

[19] S. Lachapelle, P. Brouillard, T. Deleu, and S. L. Julien. Gradient-based neural DAG learning. *CoRR*, abs/1906.02226, 2019.

[20] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania, and S. Chintala. Pytorch distributed: Experiences on accelerating data parallel training. *CoRR*, abs/2006.15704, 2020.

[21] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables, 2017.

[22] E. Nalisnick and P. Smyth. Stick-breaking variational autoencoders, 2017.

[23] I. Ng, Z. Fang, S. Zhu, Z. Chen, and J. Wang. Masked gradient-based causal structure learning, 2020.

[24] J. Pearl. Causality: Models, reasoning, and inference, second edition. *Causality*, 29, 01 2000.

[25] J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference - Foundations and Learning Algorithms*. Adaptive Computation and Machine Learning Series. The MIT Press, Cambridge, MA, USA, 2017.

[26] M. Powell. A method for nonlinear constraints in minimization problems. *In: Fletcher, R., Ed., Optimization, Academic Press*, pages 283–298, 1969.

[27] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning (ICML)*, 2015.

[28] D. J. Rezende, S. M., and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models, 2014.

[29] 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.

[30] J. Sethuraman. A constructive definition of Dirichlet priors. 1991.

[31] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*, volume 81. 01 1993.

[32] P. Spirtes, C. Meek, and T. S. Richardson. Causal inference in the presence of latent variables and selection bias. *CoRR*, abs/1302.4983, 2013.

[33] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. *CoRR*, abs/1711.00937, 2017.

[34] T. S. Verma and J. Pearl. On the equivalence of causal models. *CoRR*, abs/1304.1108, 2013.

[35] Y. Wang, L. Solus, K. D. Yang, and C. Uhler. Permutation-based causal inference algorithms with interventions, 2017.

[36] X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing. Dags with no tears: Continuous optimization for structure learning, 2018.

[37] X. Zheng, C. Dan, B. Aragam, P. Ravikumar, and E. P. Xing. Learning sparse nonparametric dags, 2020.

A. Kullback–Leibler Divergences

$$\begin{aligned}
&D_{KL}\big[q(\tilde{\mathcal{M}}, z^{(i)})||p(\tilde{\mathcal{M}}, z^{(i)})\big] = \\
&= \sum_{k=0}^{T}\sum_{l=1}^{h} D_{KL}\big(q(u_{kl})||\mathcal{N}(0,1)\big) \\
&+ \sum_{k=0}^{T}\sum_{j=1}^{d} D_{KL}\big(q(r_{kj})||\text{Bern}(\gamma)\big) \\
&+ \sum_{k=0}^{T} D_{KL}\big(q(v_k)||p_\theta(v_k|\alpha)\big) \\
&+ \mathbb{E}_{\tilde{\mathcal{M}}, z\sim q(\tilde{\mathcal{M}}, z)}\left[\log\frac{q(z|u_0,\dots,u_T)}{p(z|\beta_0,\dots,\beta_T)}\right]
\end{aligned}$$

Stick-breaking weights

$$\begin{aligned}
&D_{KL}\big(q(v_k)||p_\theta(v_k|\alpha)\big) = \\
&\quad \log\Big(\frac{\mathrm{B}\big(\rho_k w_k, (1-\rho_k)w_k\big)}{\mathrm{B}(1,\alpha)}\Big) \\
&\quad + (1-\rho_k w_k)\psi(1) \\
&\quad + \big(\alpha - (1-\rho_k)w_k\big)\psi(\alpha) \\
&\quad + (-1 + w_k - \alpha)\psi(1+\alpha)
\end{aligned}$$

Intervention embeddings

$$\begin{aligned}
&D_{KL}\big(q(u_k l)||\mathcal{N}(0,1)\big) = \\
&\quad = \frac{\sigma_{kl}^2 + \mu_{kl}^2 - 1}{2} - \log\sigma_{kl}
\end{aligned}$$

Intervention targets

$$\begin{aligned}
&D_{KL}\big(q(r_{kj})||\text{Bernoulli}(\gamma)\big) = \\
&\quad \pi_{kj}\Big(\text{logit}(\pi_{kj}) - \gamma\Big) + \log\frac{1-\pi_{kj}}{1-\sigma(\gamma)}
\end{aligned}$$

Correspondence

$$\begin{aligned}
&\mathbb{E}_{\mathcal{V}^k, u_0,\dots,u_K\sim q(\mathcal{V}^k)q(u_0)\dots q(u_K)}\Big[\\
&D_{KL}\big(q(z)||p_\theta(z|\beta_0,\beta_1,\dots,\beta_K)\big)\Big] = \\
&= \sum_{k=0}^{K}\mathbb{E}_{u_k\sim q(u_k)}\Big[\\
&q(z_k)\big(\log q(z_k) - \mathbb{E}_{\mathcal{V}^k\sim q(\mathcal{V}^k)}\left[\log\beta_k\right]\big)\Big]\Big]
\end{aligned}$$

where

$$\begin{aligned}
&\mathbb{E}_{\mathcal{V}^k\sim q(\mathcal{V}^k)}[\log\beta_k] = \\
&\qquad \mathbb{E}_{v_k\sim q(v_k)}[\log v_k] + \\
&\qquad \sum_{k'=0}^{k-1}\mathbb{E}_{v_{k'}\sim q(v_{k'})}[\log 1 - v_{k'}] \\
&= \psi(\rho_k w_k) + \sum_{k'=0}^{k-1}\psi\big((1-\rho_{k'})w_{k'}\big) - \sum_{k'=0}^{k}\psi(w_{k'}),
\end{aligned}$$

and ψ is the digamma function. We approximate, in closed form, using the Taylor series expansion, where we use the reparametrization trick on u_k to estimate the expectations.